공학석사 학위논문

Development of a Drift-Flux Model Based Pinwise Whole Core Thermal-Hydraulics Analysis Code Aiming at Highly Parallelized Execution

고효율 병렬 실행을 목표로 한 Drift-flux 모델 기반 봉단위 전노심 열수력 해석코드 개발

2016년 8월

서울대학교 대학원
에너지시스템공학부
이 재 진

Development of a Drift-Flux Model Based Pinwise Whole Core Thermal-Hydraulics Analysis Code Aiming at Highly Parallelized Execution

고효율 병렬 실행을 목표로 한 Drift-flux 모델 기반 봉단위 전노심 열수력 해석코드 개발

지도교수 주 한 규

이 논문을 공학석사 학위논문으로 제출함

2016년 6월

서울대학교 대학원

에너지시스템공학부

이 재 진

이재진의 공학석사 학위논문을 인준함

2016년 6월

위 원 장 조 형 규 (인)

부위원장 주 한 규 (인)

위 원 김 응 수 (인)

Abstract

In order to improve the parallel computation efficiency of neutronics/thermal-hydraulics (T/H) coupled reactor core calculations, a core T/H analysis code ESCOT, that can handle pinwise flow channels in the whole core calculation, is developed based on the drift-flux model and a SIMPLE-like numerical solution scheme. The governing equations are formulated and discretized from a three-dimensional 4-equation model to derive the pressure equation coupled with the equations of scalar variables.

The initial verification and validation are performed for single-phase flow conditions to assure the accuracy of the code. The calculated results are comparing with the analytic solutions, experiments, and the results of other codes such as CUPID, CTF and MATRA. The selected problems deal with the following phenomena: pressure drop by gravity acceleration and spacer grids, turbulent mixing, crossflow by friction-flow-split and asymmetric flow inlet, reverse flow by recirculation, and simplified main steam line break (MSLB) accident. It turns out that ESCOT is about 3 times faster than CTF while retaining comparable accuracy.

In order to establish an effective linear solver for the pressure equation on parallel computing platforms, the efficiency of various linear solvers is examined. The selected linear solvers are a direct solver, SuperLU, and two Krylov subspace algorithms, GMRES and BiCGSTAB. The BILU3D preconditioner is applied to accelerate the Krylov subspace algorithms, and the Krylov subspace calculation modules are parallelized with OpenMP. The incomplete domain decomposition is applied to forward and backward substitutions to solve the preconditioner equation in parallel. Parallel performance tests are carried out with sample problems, and it is shown that the unpreconditioned BiCGSTAB yields the best performance in terms

of computing time and speedup.

Keyword : Drift-flux model
SIMPLE algorithm
Pinwise core thermal-hydraulics analysis
Krylov Subspace method
Parallel execution
Student Number : 2014-20533

Contents

List of Figures

List of Tables

Chapter 1. Introduction

1.1. Background

The thermal-hydraulic (T/H) analysis of the core of a Light Water Reactor (LWR) is important in the neutronics point of view as well as in the safety points of view. It is because the T/H information is required to incorporate thermal-feedback at both steady-state and transient states. The thermal feedback effect which is through the temperature-dependent macroscopic cross sections affects significantly the power profile of the reactor, and the power profile influences the temperature profile vice versa. Thus obtaining the mutually affecting temperature and power profiles with sufficient detail becomes a crucial task in the multi-physics coupled high-fidelity reactor simulation nowadays.

Traditionally, detailed T/H analyses were performed mostly to assess the Departure Nucleate Boiling Ratios (DNBRs). The peak fuel temperature and peak cladding temperature were also obtained as the results of the core T/H analysis to determine the safety margin of the reactor core. On the other hand, the two-phase flow analysis is essential in the simulation of the accidents in Pressurized Water reactors (PWRs) and to analyze Boiling Water Reactors (BWRs) in steady states.

Because of these needs, many subchannel T/H analysis codes were developed with various two-phase models. Moreover, some of them are used in the high-fidelity multiphysics coupled core calculation, which becomes more practical with the advent of high-performance computing technology. The COBRA-TF (CTF) subchannel code[1], which is based on the two-fluid, three-field model, had been coupled with numerous neutron transport codes such as MPACT[2], TORT-TD[3], and nTRACER.[4,5,6,7] The boron tracking model implemented in CTF helps the multi-physics analyses. The effect of asymmetric boron inlet to the power profile was analyzed by nTRACER/CTF.[7] The MPACT/CTF/MAMBA system developed by the Consortium for Advanced Simulation of Light Water Reactors (CASL) predicted the CRUD induced axial power shift at an end of cycle by estimating the boron decomposition due to CRUD.[8] Nevertheless, the calculation efficiency of such coupled codes is not satisfactory because of the heavy computational burden of CTF.

The MATRA subchannel analysis code[9] of KAERI which is based on the Homogeneous Equilibrium Model (HEM) was coupled with many neutronics codes as well such as MATSER[10], DeCART[11], and nTRACER.[12,13] The effect of crossflow

in neutron transport calculations was analyzed by nTRACER/MATRA.[13] The choice of HEM and the marching scheme in MATRA results in the significant reduction of the computational burden. However, the applicability of the HEM to the two-phase flow conditions is limited to the normal operating conditions or benign upset conditions of PWRs. The MATRA code relaxes the limitation by adopting the slip model, but it is desired to employ a better model which has a wider applicability than the HEM in order to make the code applicable to the core states involving considerable boiling.

1.2. Purpose and Scope

The ultimate purpose of this research is to develop a sufficiently accurate pinwise core T/H analysis code which can cover at least two flow regimes of two-phase flow including the bubbly flow regime, and also can be executed efficiently on massively parallel computing platforms so that the code can perform high-fidelity multiphysics analysis of the LWR cores. Developing such an accurate and efficient pinwise core T/H code can relieve the computational burden of neutronics-T/H coupled calculations. The reduction in the execution time in the coupled code can make the multiphysics core analyses more affordable and it would enable realistic core T/H simulation. Eventually, more precise prediction of the safety margins becomes possible which in turn would yield economic benefits by alleviating the conservatism in the designs.

It is aimed to make the newly developed code to be applicable to multiphysics core analyses of not only PWRs, but also BWRs. The goal of this current research is, however, to develop the initial version of the code which is applicable to single-phase flow analyses of PWRs. The new pinwise core T/H analysis code is named ESCOT which is the acronym of Efficient Simulator of Core Thermal-hydraulics.

The drift-flux model (DFM)[14] is chosen here as the base T/H formulation, which would deliver sufficient accuracy with considerable simplifications because it is a good compromise between the two-fluid model and the HEM. Examples of the previous DFM based subchannel codes are WOSUB[15] and ASSERT[16]. The system analysis codes such as THOR[17], TASS/SMR[18], and RETRAN-3D[19] incorporate the DFM as one of the solver options.

The SIMPLE[20]-like algorithms are selected as the underlying numerical scheme to solve the pressure-velocity coupled problems since SIMPLE has the advantage over

the ICE[21] or SMAC[22] algorithm by employing relatively long time steps due to its implicit nature in the treatment of velocity. It is expected that taking the pressure-velocity coupled algorithm rather than the marching scheme would render benefits in parallelization because the pressure-velocity coupled algorithm solves the pressure equation over the whole domain rather than plane-wise subdomains.

Although the solution accuracy of ESCOT should be guaranteed, this research is more concerned about the computing efficiency. In particular, the solution of the linear system involving the pressure equation was tried with numerous solvers because it is well known that the convergence rate of the iterative solvers for the pressure equation is low. Many researches tried to find a proper linear solver for the pressure equation, and it turned out that the Krylov Subspace Methods (KSM) can be one of the best choices[23,24]. Due to this reason, a direct solver, SuperLU[25], and two KSMs, GMRES[26] and BiCGSTAB[27], are implemented into ESCOT. Since the convergence rate of the KSMs can be improved by adopting proper preconditioning schemes, the Blockwise Incomplete LU factorization in 3D (BILU3D)[28] is applied as a preconditioner of the KSMs. It was based on the observation that a considerable execution time reduction could be achieved by applying the BILU3D preconditioned BiCGSTAB in solving the linearized continuity equation.[23] The efficiency of various linear solvers is examined with test problems.

Since parallelization is essential in order to deal with large-scale problems, the operations in KSMs are parallelized with OpenMP as an initial step. The operations involving BILU preconditioner, however, cannot be easily parallelized, because they consist of forward and backward substitutions which require serial calculations. To resolve this problem, two domain decomposition methods are adopted; one is an approximation by ignoring the coupling with neighboring subdomains[29], and the other is an approximation by obtaining solutions with diagonal elements in a matrix[30]. The parallel efficiency of various linear solvers with or without those domain decomposition methods is investigated.

1.3. Outline of the Thesis

In Chapter 2, the theoretical background and the numerical algorithm of ESCOT are presented. The DFM based four equations and the constitutive relations are presented with the discretized form of the field equations. The pressure-velocity coupled algorithm is then described.

In Chapter 3, the construction of ESCOT is presented, and the verification and

validation of the code's solution are performed by comparing the ESCOT solutions with the analytic solutions, experimental results, and other codes' results. Then the efficiency of various linear system solvers including the KSMs is examined in Chapter 4. Chapter 5 deals with the parallelization involving the domain decomposition of preconditioner. The parallel performance of ESCOT is examined with a set of test problems. Chapter 6 concludes the thesis.

Chapter 2. Formulation of Numerical Solution Scheme for Drift-flux Model

2.1. Drift-flux Model

The drift-flux model proposed firstly by Zuber and Findlay in 1965[14] is one of the two-phase models. The basic simplification of the drift-flux model is to use the mixture velocity in the momentum conservation equation rather than using the gas and liquid velocities, individually. This simplification leads to a great advantage that can reduce the computational burden by establishing so-called “5 equations” model. It is in contrast to the two-fluid model that involves 6 equations model or even 9 equations in the two-fluid, three-field model. Moreover this approach can avoid to model the interfacial area constitutive relations, one of the most complex and difficult parts in the two-fluid models[31].

The mixture velocity in the drift-flux model can be separated into each phase velocity by adopting two drift-flux parameters, namely the distribution parameter and the drift velocity. Those parameters can be determined by experimental results, and this is the reason why the drift-flux model is a semi-empirical method. By using these parameters, one can model the vapor-liquid velocity slip, which cannot be considered in the standard HEM. The drift-flux model is known as a good practical model for the bubbly, slug, churn, and counter-current flow[24]. In the following section, the derivation of the drift-flux parameters and their physical meaning are presented.

2.1.1. Derivation of drift-flux parameters

From the vapor phase velocity, the drift-flux parameters are derived from the following split:

$$u_v = j + (u_v - j)\,. \tag{2.1}$$

Performing area average after multiplying both side by void fraction yields:

$$\langle j_v \rangle = \langle \alpha j \rangle + \langle \alpha (u_v - j) \rangle \tag{2.2}$$

where the bracket signifies at flow area averaging as:

$$\langle \phi \rangle = \frac{1}{A} \int \phi dA\,. \tag{2.3}$$

Let us define the distribution parameter, C_0, and the gas drift velocity V_{gj} as follows:

$$C_0 \equiv \frac{\langle \alpha j \rangle}{\langle \alpha \rangle \langle j \rangle} \tag{2.4}$$

$$V_{gj} \equiv \frac{\langle \alpha (u_g - j) \rangle}{\langle \alpha \rangle}. \tag{2.5}$$

Thus, Eq.(2.2) becomes as follows:

$$\langle j_v \rangle = \langle \alpha \rangle \langle j \rangle C_0 + \langle \alpha \rangle V_{gj}. \tag{2.6}$$

2.1.2. Physical meaning of the drift-flux parameters

The distribution parameter represents how the void profile appears across the flow channel[32]. If a void fraction profile is uniform, then C_0 is equal to 1. When the void fraction profile is a convex shape to the center, C_0 is greater than 1, whereas it is less than 1 when the profile is a concave shape. As presented in Figure 2-1, the distribution parameter is closely related with the two-phase flow regime. As the boiling starts, small bubbles are generated near the wall (C_0<1). When more bubbles are generated and two-phase flow is developed, the bubbles tend to move toward the center of the channel (C_0>1). The two-phase flow is turned into the pure vapor flow in the end, and uniform void fraction is made up (C_0=1). On the other hand, the gas drift velocity represents the effect of a relative velocity between the gas velocity and superficial velocity[32].

Those parameters are obtained empirically. The distribution parameter is affected by various factors such as flow pattern[34], flow channel geometry[35,36], flow channel size[37,38], flow orientation[39], pressure[34], liquid velocity[40], bubble size[41], gravitational acceleration[42], and phase change[43]. The gas drift velocity is affected by listing factors: flow pattern[34], flow channel confinement[35], and flow channel size[37,38]. Therefore, a proper correlation has to be selected to predict a right void fraction in the system of interest.

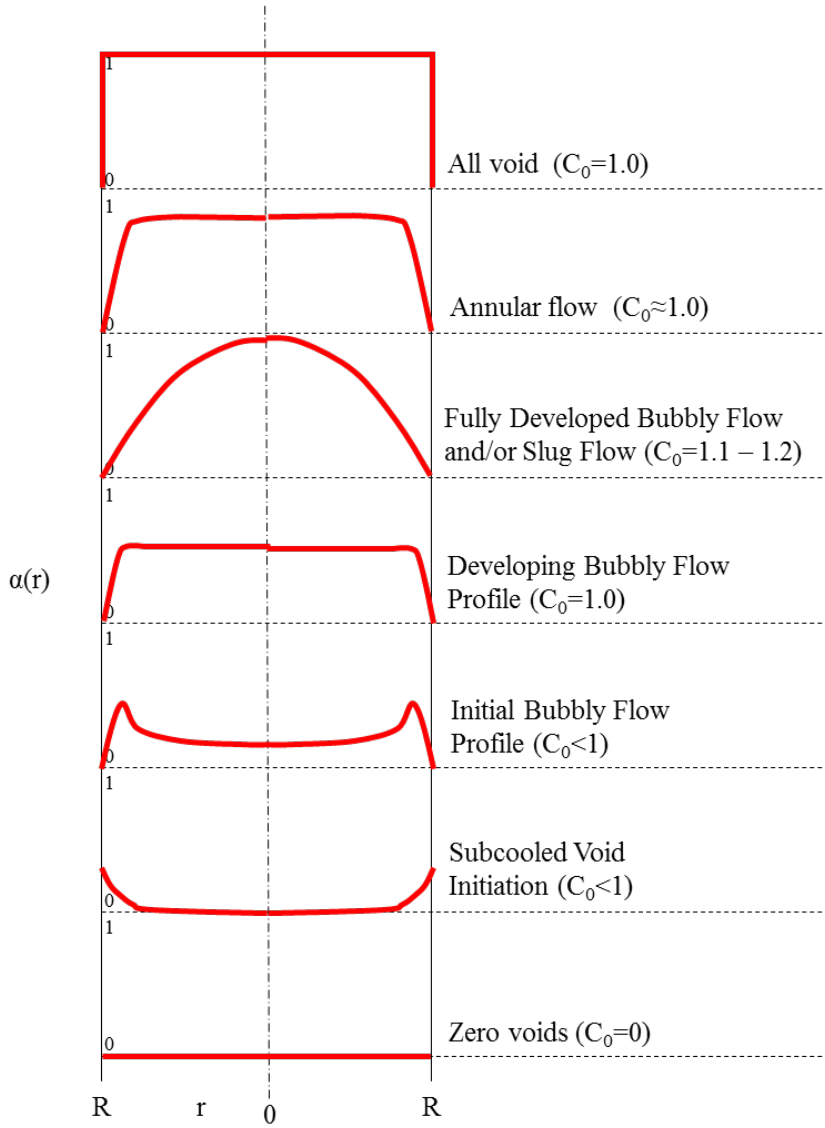

Figure 2-1. Distribution parameter by void fraction profiles for selected flow regimes as presented by Lahey (1974)[33]

2.1.3. Useful relations for deriving the drift-flux based field equations

To formulate governing equations, some relations have to be defined. First, the mixture properties are defined by[34]:

mixture density as:

$$\rho_m = \alpha\rho_v + (1-\alpha)\rho_l\,, \tag{2.7}$$

mixture velocity as:

$$\vec{u}_m = \frac{\alpha\rho_v\vec{u}_v + (1-\alpha)\rho_l\vec{u}_l}{\rho_m}, \tag{2.8}$$

and mixture enthalpy as:

$$h_m = \frac{\alpha\rho_v h_v + (1-\alpha)\rho_l h_l}{\rho_m}. \tag{2.9}$$

Secondly, the slip velocity is defined as:

$$u_r = \langle u_v \rangle - \langle u_l \rangle\,. \tag{2.10}$$

and the relation between the slip velocity and drift flux parameters can be derived as the following by using Eqs.(2.6) and (2.10):

$$u_r = \frac{\langle j \rangle (C_0 - 1) + V_{gj}}{(1-\langle\alpha\rangle)} = \frac{V'_{gj}}{(1-\langle\alpha\rangle)}\,. \tag{2.11}$$

The final relations are for the vapor and liquid velocities expressed in terms of the mixture velocity and the drift-flux parameters. Starting from Eq.(2.8) and by replacing each phase velocity with Eq.(2.11), the following relations can be derived as follows:

$$u_l = u_m - \frac{\alpha}{1-\alpha}\frac{\rho_v}{\rho_m}V'_{gj} \tag{2.12}$$

$$u_v = u_m + \frac{\rho_l}{\rho_m}V'_{gj}\,. \tag{2.13}$$

2.2. Field Equations

2.2.1. Mixture mass conservation equation

The generalized phasic mass conservations are given below for each phase:

$$\frac{\partial \alpha \rho_v}{\partial t} + \nabla \cdot (\alpha \rho_v \vec{u}_v) = \Gamma_v \quad \text{(for vapor)} \tag{2.14}$$

and

$$\frac{\partial}{\partial t}[(1-\alpha)\rho_l] + \nabla \cdot ((1-\alpha)\rho_l \vec{u}_l) = -\Gamma_v \quad \text{(for liquid)} \tag{2.15}$$

where Γ_v is the volumetric vapor generation rate.

By summing up the two equations and also by using Eqs.(2.7) and (2.8), the mixture mass conservation equation is obtained as follows:

$$\frac{\partial \rho_m}{\partial t} + \nabla \cdot (\rho_m \vec{u}_m) = 0 \ . \tag{2.16}$$

2.2.2. Vapor mass conservation equation

By inserting the vapor velocity given by Eq.(2.13) into Eq.(2.14), the vapor mass conservation equation in the drift-flux model can be defined as follow:

$$\frac{\partial \alpha \rho_v}{\partial t} + \nabla \cdot \left(\alpha \rho_v \vec{u}_m\right) = \Gamma_v - \nabla \cdot \left(\frac{\alpha \rho_l \rho_v}{\rho_m} \vec{V}'_{gj} \right). \tag{2.17}$$

Note that the divergence term with the gas drift velocity is unique, which only appears at the field equations of two-phase flow when the drift-flux model is applied.

2.2.3. Mixture momentum conservation equation

The generalized phasic momentum equation is given by Eq.(2.18):

$$\begin{aligned} &\frac{\partial}{\partial t}\left(\alpha_k \rho_k \vec{u}_k\right) + \nabla \cdot \left(\alpha_k \rho_k \vec{u}_k \vec{u}_k\right) \\ &\quad = \alpha_k \rho_k \vec{g} - \alpha_k \nabla P + \nabla \cdot \left[\alpha_k \left(\boldsymbol{\tau}_k^{ij} + \mathbf{T}_k^{ij} \right) \right] + \vec{M}_k^L + \vec{M}_k^d + \vec{M}_k^T \end{aligned} \tag{2.18}$$

where $\boldsymbol{\tau}_k^{ij}$: viscous shear stress

$\mathbf{T}_k^{ij}$: Turbulent shear stress

$\vec{M}_k^L$: Momentum source or sink due to phase change

$\vec{M}_k^d$: Interfacial drag force

$\vec{M}_k^T$: Momentum exchange due to turbulent mixing.

Here the subscript *k* denotes phase. The turbulent shear stress term is not modeled in the code so that it will be eliminated in future formulations. Because the drift-flux model deals with the mixture velocity, the phase velocities should be replaced with the mixture velocity. By summing the two phase momentum equations to eliminate $\vec{M}_k^L$ and $\vec{M}_k^d$, the mixture momentum conservation equation can be established by using Eqs.(2.7), (2.8), (2.12) and (2.13) as:

$$\begin{aligned}\frac{\partial}{\partial t}\left(\rho_m \vec{u}_m\right)+\nabla\cdot\left(\rho_m \vec{u}_m \vec{u}_m\right)&=\rho_m \vec{g}-\nabla P\\&+\nabla\cdot\boldsymbol{\tau}_m^{ij}+\vec{M}_m^T-\nabla\cdot\left(\frac{\alpha}{1-\alpha}\frac{\rho_v\rho_l}{\rho_m}\vec{V}_{gj}'\vec{V}_{gj}'\right).\end{aligned} \tag{2.19}$$

2.2.4. Mixture energy conservation equation

There are several types of energy conservation equation. Here, the energy conservation equation using enthalpy as a primary variable is used. The following assumptions are applied to derive the energy conservation equation:

1) Kinetic energy is negligible as compared with internal energy.
2) Heat transfer by the heat conduction among fluid is negligible.

$$\begin{aligned}\frac{\partial}{\partial t}\left(\alpha_k\rho_k h_k\right)+\nabla\cdot\left(\alpha_k\rho_k h_k\vec{u}_k\right)&=\\-\nabla\cdot\left[\alpha_k\vec{Q}_k''^T\right]+\Gamma_k h_k^i+q_{wk}'''+q_{vk}'''+\alpha_k\frac{\partial P}{\partial t}&\end{aligned} \tag{2.20}$$

where $\vec{Q}_k''^T$: Turbulence heat flux

$\Gamma_k h_k^i$: Energy transfer to phase change

q_{wk}''' : Volumetric wall heat transfer rate

q_{vk}''' : Volumetric heat generation rate.

By summing up the two phase equations and by replacing the phase parameters with mixture ones based on Eqs.(2.9), (2.12) and (2.13), the mixture energy conservation in the DFM can be derived as follow:

$$\frac{\partial}{\partial t}\left(\rho_m h_m\right)+\nabla\cdot\left(\rho_m h_m \vec{u}_m\right)= \\ -\nabla\cdot\vec{Q}_m''^{T}+q_w'''+q_v'''+\frac{\partial P}{\partial t}-\nabla\cdot\left(\frac{\alpha\rho_v\rho_l}{\rho_m}\left(h_v-h_l\right)\vec{V}_{jg}'\right) . \tag{2.21}$$

In fact, one more equation is necessary to close the 5 equations model in the DFM. However, for simplicity, it is assumed that the state of vapor is saturated. Therefore, the 4 equation model will be used in the code.

2.3. Constitutive Relations

2.3.1. Equation of state

Four primary variables will be selected to solve the 4 equations. Those will be the mixture velocity, void fraction, pressure, and mixture enthalpy. The other variables are designated as the secondary variables, which can be determined by using the equation of state or simply steam table (IAPWS-IF97).

The mixture density can be calculated with each phase density and void fraction by Eq.(2.7). The vapor and liquid density can be calculated by the steam table as the function of pressure and enthalpy as:

$$\rho_l=\rho_l(P,h_l) \ . \tag{2.22}$$

And the vapor density is a function of only pressure because the saturated state is assumed:

$$\rho_v=\rho_{v,sat}(P) \ . \tag{2.23}$$

The phasic temperatures can be obtained by the same manner as the phasic densities:

$$T_l=T_l(P,h_l) \tag{2.24}$$

and

$$T_v=T_{v,sat}(P) \ . \tag{2.25}$$

Mixture enthalpy is a function of density and enthalpy of each phase and void fraction as well. Originally, each enthalpy should be calculated from two energy conservation equations, but because vapor is assumed to be saturated, only liquid

enthalpy is obtained from Eq.(2.21), and vapor enthalpy is calculated from the steam table as a function of pressure as below:.

$$h_v = h_{v,sat}(P). \tag{2.26}$$

2.3.2. Interfacial mass transfer

The vapor mass conservation equation takes into account the vaper generation term which can be a sum of two terms: the volumetric mass transfer rate between the interface of vapor and liquid and the volumetric vapor generation rate from the thermal boundary layer near the wall. The wall model is not considered yet. So if saturated vapor is assumed, the vapor generation term can be defined as follows:

$$\begin{aligned}\Gamma_v &= \Gamma_{iv} \\ &= \frac{H_{iv}(T_v - T_{sat}) + H_{il}(T_l - T_{sat})}{h_{vi} - h_{li}} = \frac{H_{il}(T_l - T_{sat})}{h_{v,sat} - h_{li}}\end{aligned} \tag{2.27}$$

where $h_{li} = \begin{cases} h_l & \text{if } H_{il} \geq 0 \\ h_{l,sat} & \text{if } H_{il} < 0 \end{cases}$

$H_{il} = 2.78\,\text{BTU/ft}^2 \cdot \text{sec} \cdot {}^{\circ}\text{F}$ for the small bubble regime and subcooled vapor[1].

2.3.3. Viscous shear stress

The viscous shear stress contains the friction loss and form loss as:

$$\nabla \cdot \boldsymbol{\tau}_m^{ij} = -\left(\left.\frac{dP}{dX}\right|_{friction} + \left.\frac{dP}{dX}\right|_{form} \right). \tag{2.28}$$

The friction loss is only applied to the axial momentum as follows:

$$\left.\frac{dP}{dX}\right|_{friction} = \frac{f}{2D_h}\rho_m w_m \left| w_m \right|. \tag{2.29}$$

For the laminar flow (Re <2,200), the analytic solution for the friction factor is applied:

$$f = \frac{64}{\text{Re}}. \tag{2.30}$$

For the flow which has Re larger than 3,000, two correlations can be used as shown in Table 2-1. The correlation depends on the geometry of the problem so that it can

be changed by users' choice.

Table 2-1. Correlations of the friction factor in the turbulent flow region[1]

Correlation	Reference
$f = 0.204\,\mathrm{Re}^{-0.2}$	McAdam's correlation
$\frac{1}{\sqrt{f}} = -2.00\log\left(\frac{\varepsilon}{3.7D} + \frac{2.51}{\mathrm{Re}\sqrt{f}}\left[1.14 - 2\log\left(\frac{\varepsilon}{D} + \frac{21.25}{\mathrm{Re}^{0.9}}\right)\right]\right)$	Zigrang-Slyvester correlation

The form loss is applied in the both axial and lateral momentum equations as follows:

$$\left.\frac{dP}{dX}\right|_{form} = \frac{K_x}{2\Delta x}\rho_m u_m \left|u_m\right|. \tag{2.31}$$

The form loss in the axial direction might appear due to grid spacers and the form loss in the lateral direction indicates the resistance of lateral flow that occurs when it passes a gap between two rods. The form loss factor should be given by the user.

2.3.4. Turbulent mixing models

The mixing phenomenon by turbulent flow can be taken into account in various ways. In many subchannel codes, turbulent mixing models are used to simulate turbulent mixing effects. As stated in Kelly *et al.*[44], the turbulent mixing and void drift in a mass of vapor phase can be formulated as follow:

$$W_{tv} = \nabla \cdot \vec{W}''_{tv} = \sum \frac{s_{ij}}{A_i}\left(W''_{tv}\right)_{ij} \tag{2.32}$$

where s_{ij} : Gap between channel i and j

A_i : Flow area of channel i.

And the flux term is defined as below:

$$\left(W''_{tv}\right)_{ij} = \frac{\varepsilon}{l}\left[\left(\alpha\rho_v\right)_i - \left(\alpha\rho_v\right)_j - \left[\left(\alpha\rho_v\right)_i - \left(\alpha\rho_v\right)_j\right]_{FD}\right] \tag{2.33}$$

where ε: Eddy diffusivity (m^2/sec)

l: Effective mixing length (m).

This is the equal-volume-exchange model which states that the quantity of fluid is exchanged with that of neighboring subchannels in equal volumes. Note that this works only in the lateral direction. In the axial direction, the convection term is dominant usually so that the effect is negligible. The first difference term represents the mixing effect between channel *i* and *j* by turbulent flow and the second difference term does mixing between channel *i* and *j* by the void drift. The subscript FD in void mixing denotes fully developed. Here, void drift mixing is not modeled yet, but only turbulent mixing is considered. In analogy with the vapor mass mixing term, the momentum and energy mixing terms can be defined as:
turbulent mixing in the mixture momentum is:

$$M_m^T = \sum_j \frac{s_{ij}}{A_i}\left(\frac{\varepsilon}{l}\right)\left[\left(\rho_m v_m\right)_j - \left(\rho_m v_m\right)_i\right] \tag{2.34}$$

and turbulent mixing in the mixture energy is:

$$-\nabla\cdot\vec{Q}_m''^T = \sum_j \frac{s_{ij}}{A_i}\left(\frac{\varepsilon}{l}\right)\left[\left(\rho_m h_m\right)_j - \left(\rho_m h_m\right)_i\right]. \tag{2.35}$$

Now it is necessary to determine the mixing coefficient which consists of eddy diffusivity and effective mixing length. The mixing coefficient is related with the mixing parameter, β, averaged density, and the axial mass flux as follows[1]:

$$\frac{\varepsilon}{l} = \frac{\beta\bar{G}}{\bar{\rho}} \tag{2.36}$$

where $\bar{G}$: Averaged axial mass flux between adjacent channels [kg/m^{2}sec]
$\bar{\rho}$: Averaged density between adjacent channels [kg/m^3].

The physical meaning of β is the ratio of the transverse mass flux to the axial mass flux. β can be calculated from the correlation suggested by Roger & Rosenhard in 1972[45].

$$\beta = \begin{cases} 0.5\lambda\,\mathrm{Re}^{-0.1}\left[1+\left(\frac{D_{h,j}}{D_{h,j}}\right)^{1.5}\right]\left(\frac{D_{h,i}}{D_{rod}}\right) & \text{if } D_{h,i} < D_{h,j} \\ 0.5\lambda\,\mathrm{Re}^{-0.1}\left[1+\left(\frac{D_{h,i}}{D_{h,j}}\right)^{1.5}\right]\left(\frac{D_{h,j}}{D_{rod}}\right) & \text{if } D_{h,i} > D_{h,j} \end{cases} \tag{2.37}$$

where $\lambda = 0.0058\left(\frac{s_{ij}}{D_{rod}}\right)^{-1.46}$. However, in most cases β is chosen by the users' input.

2.3.5. Drift-flux parameters

The Chexal-Lellouche correlation (1991)[46] is implemented into the code to calculate the drift-flux parameters. For the distribution parameter:

$$C_0 = \frac{L}{K_0 + (1 - K_0)\alpha^{\gamma}} \tag{2.38}$$

where $L = \frac{L_n}{L_d}$

$$L_n = \begin{cases} 1 - \exp(-\alpha C_1) & \text{if } \alpha C_1 < 170 \\ 1 & \text{otherwise} \end{cases}$$

$$L_d = \begin{cases} 1 - \exp(-C_1) & \text{if } C_1 < 170 \\ 1 & \text{otherwise} \end{cases}$$

$$C_1 = \frac{4P_{crit}^2}{P(P_{crit} - P)}$$

$$K_0 = B_1 + (1 - B_1)\left(\frac{\rho_v}{\rho_l}\right)^{0.25}$$

$$B_1 = \min\left(0.80, \frac{1}{1 + \exp(-\max(\mathrm{Re}_f, \mathrm{Re}_v)/6\times10^4)}\right)$$

$$\gamma = \frac{1 + 1.57(\rho_v / \rho_l)}{1 - B_1}$$

.

For the gas drift velocity:

$$1.41\left[\frac{g\sigma(\rho_l - \rho_v)}{\rho_l^2}\right]^{0.25} C_2 C_3 C_4 C_5 \tag{2.39}$$

where $C_2 = \begin{cases} 0.4757\left(\ln\dfrac{\rho_l}{\rho_v}\right)^{0.7} & \text{if } \dfrac{\rho_l}{\rho_v} \le 18 \\ 1 & \text{if } C_6 \ge 1 \\ \left(1-\exp\left(\dfrac{-C_6}{1-C_6}\right)\right)^{-1} & \text{if } C_6 < 1 \text{ and if } \dfrac{\rho_l}{\rho_v} > 18 \end{cases}$

$$C_3 = \max\left[0.5,\ 2\exp\left(\frac{-\left|\mathrm{Re}_f\right|}{6\times10^4}\right)\right]$$

$$C_4 = \begin{cases} 1 & \text{if } C_7 \ge 1 \\ \left(1-\exp\left(\dfrac{-C_7}{1-C_7}\right)\right)^{-1} & \text{if } C_7 < 1 \end{cases}$$

$$C_5 = \left(1-\alpha\right)^{B_1}$$

$$C_6 = \sqrt{150\frac{\rho_l}{\rho_v}}$$

$$C_7 = \left(\frac{0.09114}{D_h}\right)^{0.6}.$$

The Chexal-Lellouche correlation has the following features[47]:

1) Applicable to all flow regimes so that it does not need a flow regime map.
2) Applicable to all channel inclinations (i.e. vertical, horizontal, inclined), and co-current upward, downward, and counter-current flow direction are covered.
3) Applicable over a large range of channel size, pressures and mass fluxes.
4) Numerically stable due to continuous derivative of the correlation.

RELAP5 and TASS/SMR which are the system analysis codes also utilizes the Chexal-Lellouche correlation to simulate a two-phase flow.[47,48] The applicable ranges of the correlation are[47]:

$$0.01 \le \langle\alpha\rangle \le 0.95 \tag{2.40}$$

$$\left.\begin{array}{c} 0.01 \le G \le 2{,}100\ \mathrm{kg/m^2}s \\ 0.1 \le P \le 14.5\mathrm{MPa} \end{array}\right\} \text{for diabatic tests} \tag{2.41}$$

$$\left.\begin{array}{c} 0.01 \le G \le 2{,}550\ \mathrm{kg/m^2}s \\ 0.1 \le P \le 18\mathrm{MPa} \end{array}\right\} \text{for adiabatic tests}. \tag{2.42}$$

2.4. Discretization

To solve the above field equations numerically, the problem domain should be discretized to local control volumes. In many subchannel analysis codes, the control volume (CV) is defined at the subchannel or pin level. Figure 2-2 shows how the control volume of a unit cell is defined.

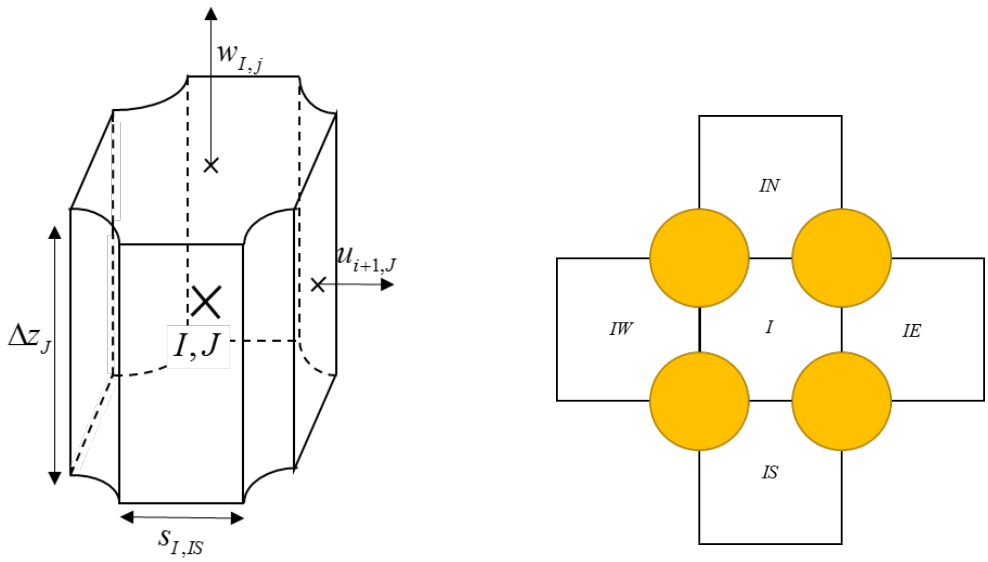

Figure 2-2. Control volume of a scalar cell at channel I and level J

In the discretized cell, the scalar variables such as density, enthalpy, pressure, and temperature are defined at the center. For velocities, there are two choices: the collocated grid based one and the staggered grid based one. If the collocated grid is adopted, then the velocity will be defined at the same CV of the scalar. The collocated grid has two advantages: First, no need to define additional CVs for the velocity and secondly, it is convenient to deal with unstructured mesh systems[49]. However, because it is hard to capture the pressure oscillation without a special treatment, it might yield wrong solutions in solving the momentum equation. Since the structured systems are anticipated in the core thermal-hydraulics calculation, the staggered grid is chosen here and the velocity is defined in momentum CVs.

The CV for the lateral momentum is illustrated at Figure 2-3. The lateral velocity is defined at the center of the CV and the scalar values are located in the surface. The CVs of the lateral momentum and the scalar variables are defined at a same axial level.

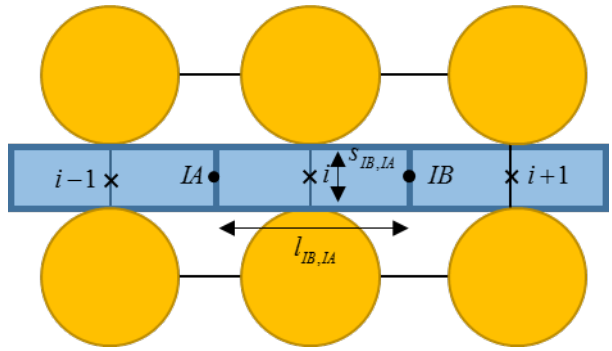

Figure 2-3. Control volume for the lateral momentum in the staggered grid

The CV for the axial momentum has the same shape as that of the scalar variables, but the only difference is that they are defined in different axial level as shown in Figure 2-4.

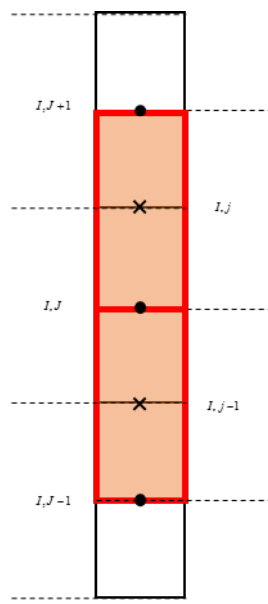

Figure 2-4. Control volume for the axial momentum in the staggered grid

The Finite Volume Method (FVM) is applied within the CV to discretize the field equations. The temporal discretization is based on the semi-implicit scheme in the first order. It means the sonic propagation and interphase interchange such as pressure and interfacial temperature which have short time scale are treated implicitly, and the fluid convections which have relatively large time scale are treated explicitly[50].

In this thesis, the variables in different CVs are distinguishable by upper and lower cases of subscripts. Variables at the center of the scalar CVs have capital letter subscripts (i.e. $\rho_{I,J}$). Variables at the center of the momentum CVs have lower letter indices at a staggered point. For example, the lateral velocity u is defined at index (i,J), and the axial velocity w has index (I,j). Sometimes, because of the surface integral by the FVM, some quantities have to be defined at surfaces of CVs. Two approaches are used to determine surface values. One is the donor cell method which determines surface variables by a characteristic of convection. Upstream variables are taken by a sign of a velocity at a surface. The donor cell scheme in first order can be expressed as follows:
For a radial direction,

$$\phi_{i,J}^{\bar{n}} = \begin{cases} \phi_{IA,J}^{n} & \text{if } u_{i,J}^{n} \geq 0 \\ \phi_{IB,J}^{n} & \text{if } u_{i,J}^{n} < 0 \end{cases} \tag{2.43}$$

and in case of an axial direction,

$$\phi_{I,j}^{\bar{n}} = \begin{cases} \phi_{I,J}^{n} & \text{if } w_{I,j}^{n} \geq 0 \\ \phi_{I,J+1}^{n} & \text{if } w_{I,j}^{n} < 0 \end{cases}. \tag{2.44}$$

The variables in which the donor cell scheme is applied have superscript $\bar{n}$.

The other is the harmonic (or inverse distance) average. This average scheme is applied to any old time step value located at a surface in CVs. Table 2-2 shows how the surface value in each CV can be defined by the harmonic average.

2.4.1. Mixture mass continuity

Performing volume integral and semi-implicit discretization to Eq. (2.16) can yield:

$$\int_{CV}\left[\frac{\partial \rho_m}{\partial t} + \nabla\cdot\left(\rho_m \vec{u}_m\right)\right]dV = \int_{CV}\frac{\partial \rho_m}{\partial t}dV + \int_{S}\left(\rho_m^{\bar{n}}\vec{u}_m^{n+1}\right)\cdot d\vec{S} = 0. \tag{2.45}$$

The discretized subchannel form of the mixture mass continuity is obtained as:

$$\begin{aligned}\frac{\rho_{m,I,J}^{n+1} - \rho_{m,I,J}^{n}}{\Delta t}A_{c,I,J}\Delta z_J + \sum_{ic=1}^{nb}\left[\rho_m^{\bar{n}}\left(\vec{u}_m^{n+1}\cdot\vec{s}\right)_{ic}\Delta z_J\right]_{ic,J} \\ + \left[\rho_m^{\bar{n}} w_m^{n+1} A_c\right]_{I,j} - \left[\rho_m^{\bar{n}} w_m^{n+1} A_c\right]_{I,j-1} = 0\end{aligned}. \tag{2.46}$$

Table 2-2. Harmonic average of the variables defined in a surface

Type of CV	Type of averaged variable	Equation
Scalar CV	Averaged scalar variables	$\phi^n_{i,J} = \dfrac{l_{i,IA}\phi^n_{IB,J} + l_{IB,i}\phi^n_{IA,J}}{l_{IB,IA}}$
		$\phi^n_{I,j} = \dfrac{\Delta z_{J+1}\phi^n_{I,J} + \Delta z_J \phi^n_{I,J+1}}{\Delta z_J + \Delta z_{J+1}}$
Momentum CV	Averaged scalar variables	$\phi^n_{i,j} = \dfrac{l_{i,IA}\Delta z_{J+1}\phi^n_{IB,J} + l_{IB,i}\Delta z_{J+1}\phi^n_{IA,J} + l_{i,IA}\Delta z_J \phi^n_{IB,J+1} + l_{IB,i}\Delta z_J \phi^n_{IA,J+1}}{l_{IB,IA}(\Delta z_J + \Delta z_{J+1})}$
	Averaged vector variables	$\nu^n_{x,IB,J} = \dfrac{1}{2}\left(\nu^n_{x,i,J} + \nu^n_{x,i+1,J}\right)$
		$\nu^n_{z,I,J+1} = \dfrac{1}{2}\left(\nu^n_{z,I,j} + \nu^n_{z,I,j+1}\right)$
		$\nu^n_{x,i,j} = \dfrac{\Delta z_{J+1}\nu^n_{x,i,J} + \Delta z_J \nu^n_{x,i,J+1}}{\Delta z_J + \Delta z_{J+1}}$
		$\nu^n_{z,i,j} = \dfrac{l_{i,IA}\nu^n_{z,IB,j} + l_{IB,i}\nu^n_{z,IA,j}}{l_{IB,IA}}$

2.4.2. Vapor mass continuity equation

Applying volume integral and semi-implicit discretization to Eq.(2.17) can yield:

$$\int_{CV}\left[\frac{\partial \alpha\rho_v}{\partial t} + \nabla\cdot\left(\alpha\rho_v\vec{u}_m\right)\right]dV = \int_{CV}\left[\Gamma_g - \nabla\cdot\left(\frac{\alpha\rho_l\rho_v}{\rho_m}\vec{V}'_{gj}\right)\right]dV\,. \qquad (2.47)$$

Let us focus on the temporal term. Applying the chain rule can separate the primary variables as follows:

$$\begin{aligned}\int_{CV}\frac{\partial \alpha\rho_v}{\partial t}dV &= \int_{CV}\left[\alpha\frac{\partial \rho_v}{\partial t} + \rho_v\frac{\partial \alpha}{\partial t}\right]dV \\ &= \left[\alpha^n_{I,J}\frac{\rho^{n+1}_{v,I,J} - \rho^n_{v,I,J}}{\Delta t} + \rho^n_{v,I,J}\frac{\alpha^{n+1}_{I,J} - \alpha^n_{I,J}}{\Delta t}\right]A_{c,I,J}\Delta z_J\end{aligned}\,. \qquad (2.48)$$

The discretized subchannel form of the vapor mass continuity is obtained as:

$$\left[\alpha_{I,J}^{n}\frac{\rho_{v,I,J}^{n+1}-\rho_{v,I,J}^{n}}{\Delta t}+\rho_{v,I,J}^{n}\frac{\alpha_{I,J}^{n+1}-\alpha_{I,J}^{n}}{\Delta t}\right]A_{c,I,J}\Delta z_J$$
$$+\sum_{ic=1}^{nb}\left[\left(\alpha\rho_v\right)^{\bar{n}}\left(\vec{u}_m^{\,n+1}\cdot\vec{s}\right)\Delta z_J\right]_{ic,J}+\left[\left(\alpha\rho_v\right)^{\bar{n}}w_m^{n+1}A_c\right]_{I,j}-\left[\left(\alpha\rho_v\right)^{\bar{n}}w_m^{n+1}A_c\right]_{I,j-1} \quad (2.49)$$
$$=\frac{H_{il}^{n}(T_l^{n+1}-T_{sat}^{n+1})}{h_{v,sat}^{n}-h_{li}^{n}}A_{c,I,J}\Delta z_J-\sum_{nnb}[CV]_{nnb}$$

where
$$\sum_{nnb}[CV]_{nnb}=\sum_{ic=1}^{nb}\left[\left(\frac{\alpha\rho_l\rho_v}{\rho_m}\right)^{n}\left(\vec{V}_{gj}'\cdot\vec{s}\right)^{n}\Delta z_J\right]_{ic,J}$$
$$+\left(\frac{\alpha\rho_l\rho_v}{\rho_m}V_{gjz}'A_c\right)_{I,j}^{n}-\left(\frac{\alpha\rho_l\rho_v}{\rho_m}V_{gjz}'A_c\right)_{I,j-1}^{n}.$$

Note that the temperatures of the interface mass transfer are treated implicitly.

2.4.3. Axial momentum equation

Let us apply volume integral and semi-implicit discretization to Eq.(2.19) to yield:

$$\int_{CV}\left[\frac{\partial}{\partial t}\left(\rho_m w_m\right)+\nabla\cdot\left(\rho_m w_m\vec{u}_m\right)\right]dV$$
$$=\int_{CV}\left[-\frac{\partial P}{\partial z}+\sum_k\nabla\cdot\left(\alpha_k\tau_k^{ij}\right)-\nabla\cdot\left(\frac{\alpha}{1-\alpha}\frac{\rho_v\rho_l}{\rho_m}V_{gjz}'\vec{V}_{gj}'\right)+M_m^{T}-\rho_m g\right]dV \quad (2.50)$$

In order to derive the semi-conservative form of the total time derivative term, the chain rule and the mixture continuity equation are used. Then the left hand side of Eq.(2.50) becomes:

$$\int_{CV}\left[\frac{\partial}{\partial t}\left(\rho_m w_m\right)+\nabla\cdot\left(\rho_m w_m\vec{u}_m\right)\right]dV$$
$$=\int_{CV}\left[\rho_m\frac{\partial w_m}{\partial t}+w_m\frac{\partial\rho_m}{\partial t}+\nabla\cdot\left(\rho_m w_m\vec{u}_m\right)\right]dV \quad (2.51)$$
$$=\int_{CV}\left[\rho_m\frac{\partial w_m}{\partial t}-w_m\cdot\nabla\left(\rho_m\vec{u}_m\right)+\nabla\cdot\left(\rho_m w_m\vec{u}_m\right)\right]dV$$

Then let us apply the first order upwind scheme to the convection term to yield:

$$
\begin{aligned}
&\int_{CV} \nabla\cdot\left(\rho_m w_m \vec{u}_m\right) dV \\
&= \int_{S} \left[\left(\rho_m \vec{u}_m\right)^n w_m^{n+1} \right] \cdot d\vec{S} \\
&= \sum_{ic=1}^{nb} \left[w_m^{n+1} \left(\left(\rho_m \vec{u}_m\right)^n \cdot \vec{s} \right) \Delta z_j \right]_{ic,j} + \left[\left(\rho_m w_m\right)^n w_m^{n+1} A_c \right]_{I,J+1} - \left[\left(\rho_m w_m\right)^n w_m^{n+1} A_c \right]_{I,J} \\
&= \sum_{ic=1}^{nb} \left[\left(\vec{F}_z^{\,n} \cdot \vec{n} \right) w_m^{n+1} \right]_{ic} + F_{z,up}^n w_{m,n}^{n+1} - F_{z,dw}^n w_{m,s}^{n+1} \\
&= \sum_{ic=1}^{nb} \Big[-\max[-\vec{F}_{z,ic}^{\,n} \cdot \vec{n}, 0] w_{m,\langle ic\rangle,j}^{n+1} + \max[\vec{F}_{z,ic}^{\,n} \cdot \vec{n}, 0] w_{m,I,j}^{n+1} \Big] \\
&\qquad + \max[F_{z,up}^n, 0] w_{m,I,j}^{n+1} - \max[-F_{z,up}^n, 0] w_{m,I,j+1}^{n+1} \\
&\qquad - \left(\max[F_{z,dw}^n, 0] w_{m,I,j-1}^{n+1} - \max[-F_{z,dw}^n, 0] w_{m,I,j}^{n+1} \right) \\
&= \sum_{ic=1}^{nb} \Big[-\max[-\vec{F}_{z,ic}^{\,n} \cdot \vec{n}, 0] w_{m,\langle ic\rangle,j}^{n+1} + \left(\vec{F}_{z,ic}^{\,n} \cdot \vec{n} + \max[-\vec{F}_{z,ic}^{\,n} \cdot \vec{n}, 0] \right) w_{m,I,j}^{n+1} \Big] \\
&\qquad + \left(F_{z,up}^n + \max[-F_{z,up}^n, 0] \right) w_{m,I,j}^{n+1} - \max[-F_{z,up}^n, 0] w_{m,I,j+1}^{n+1} \\
&\qquad - \max[F_{z,dw}^n, 0] w_{m,I,j-1}^{n+1} + \left(\max[F_{z,dw}^n, 0] - F_{z,dw}^n \right) w_{m,I,j}^{n+1}
\end{aligned}
\qquad . \quad (2.52)
$$

The subscript <*ic*> denotes the index of radially neighboring scalar cell which has an interface with gap *ic*.

In case of the second term of Eq.(2.51), it can be discretized as follows:

$$
\begin{aligned}
&\int_{CV} w_m \cdot \nabla\left(\rho_m \vec{u}_m\right) dV \\
&= \left(w_m\right)_{I,j}^{n+1} \left[\sum_{ic=1}^{nb} \left[\rho_m^n \left(\vec{u}_m^{\,n} \cdot \vec{s} \right) \Delta z_j \right]_{ic,j} + \left[\left(\rho_m w_m\right)^n A_c \right]_{I,J+1} - \left[\left(\rho_m w_m\right)^n A_c \right]_{I,J} \right] \quad (2.53) \\
&= \left(w_m\right)_{I,j}^{n+1} \left[\sum_{ic=1}^{nb} \left[\vec{F}_z^{\,n} \cdot \vec{n} \right]_{ic} + F_{z,up}^n - F_{z,dw}^n \right].
\end{aligned}
$$

Because the viscous term in the axial momentum equation contains the form loss and friction loss, the discretized form of it becomes:

$$
\int_{CV} \nabla\cdot\boldsymbol{\tau}_k^{ij} dV = -\left[\frac{f}{2D_h} \rho_m w_m \left| w_m \right| + \frac{K_z}{2\Delta z_j} \rho_m w_m \left| w_m \right| \right]_{I,j}^{n} \Delta V_{I,j} \qquad (2.54)
$$

The turbulent mixing term is applied only to the lateral neighboring cells, and the velocity is treated implicitly. Applying the above assumptions can yield the discretized form of the turbulent mixing term as:

$$\begin{aligned}\int_{CV} M_m^T dV &= \int_{CV}\sum_{IC}\frac{s_{I,IC}}{A}\left(\frac{\varepsilon}{l}\right)\left[\left(\rho_m w_m\right)_{IC,j}-\left(\rho_m w_m\right)_{I,j}\right]dV \\ &= \sum_{IC}\frac{s_{I,IC}}{A_{I,j}}\left(\frac{\varepsilon}{l}\right)_{I,IC}^{n}\left[\left(\rho_m^n w_m^{n+1}\right)_{IC,j}-\left(\rho_m^n w_m^{n+1}\right)_{I,j}\right]A_{c,I,j}\Delta z_j \\ &= \sum_{IC} s_{I,IC}\left(\frac{\varepsilon}{l}\right)_{I,IC}^{n}\left[\left(\rho_m\right)_{IC,j}^{n}\left(w_m\right)_{IC,j}^{n+1}-\left(\rho_m\right)_{I,j}^{n}\left(w_m\right)_{I,j}^{n+1}\right]\Delta z_j .\end{aligned} \tag{2.55}$$

By rearranging all implicit terms of the axial velocity to LHS and the explicit source terms and the implicit pressure to RHS, the discretized subchannel form of the axial momentum equation is given as:

$$\begin{aligned}a_{z,P} w_{m,I,j}^{n+1} - \sum_{ic=1}^{nb} a_{z,ic} w_{m,\langle ic\rangle,j}^{n+1} - a_{z,DW} w_{m,I,j-1}^{n+1} - a_{z,UP} w_{m,I,j+1}^{n+1} \\ = \left[P_{I,J-1}^{n+1} - P_{I,J}^{n+1}\right]A_{c,I,j} + S_{z,I,j}\end{aligned} \tag{2.56}$$

where $a_{z,UP} = \max\left[-F_{z,up}^{n}, 0\right]$

$$a_{z,DW} = \max\left[F_{z,dw}^{n}, 0\right]$$

$$a_{z,ic} = \max\left[-\vec{F}_{z,ic}^{n}\cdot\vec{n}, 0\right] + s_{I,IC}\left(\frac{\varepsilon}{l}\right)_{I,IC}^{n}\left(\rho_m\right)_{IC,j}^{n}\Delta z_j$$

$$a_{z,P} = a_{z,UP} + a_{z,DW} + \frac{\left(\rho_m\right)_{I,j}^{n}}{\Delta t}A_{c,I,j}\Delta z_j + \sum_{ic=1}^{nb}\left[\max[-\vec{F}_{z,ic}^{n}\cdot\vec{n}, 0] + s_{I,\langle ic\rangle}\left(\frac{\varepsilon}{l}\right)_{I,\langle ic\rangle}^{n}\left(\rho_m\right)_{I,j}^{n}\Delta z_j\right]$$

$$\begin{aligned}S_{z,I,j} = &-\left[\frac{\lambda}{2D_h}\rho_m w_m\left|w_m\right| + \frac{K_z}{2\Delta z_j}\rho_m w_m\left|w_m\right|\right]_{I,j}^{n}\Delta V_{I,j} \\ &- \sum_{nnb}\left[CM\right]_{z,nnb}^{n} + \frac{\left(\rho_m\right)_{I,j}^{n}\left(w_m\right)_{I,j}^{n}}{\Delta t}A_{c,I,j}\Delta z_j - \rho_m g\Delta V_{I,j}\end{aligned}$$

$$\begin{aligned}\sum_{nbt}\left[CM\right]_{z,nbt}^{n} = &\sum_{ic=1}^{nb}\left[\frac{\alpha}{1-\alpha}\frac{\rho_v\rho_l}{\rho_m}V'_{gjz}\left(\vec{V}'_{gj}\cdot\vec{n}\right)s\Delta z\right]_{ic,j} \\ &+\left(\frac{\alpha}{1-\alpha}\frac{\rho_v\rho_l}{\rho_m}V'_{gjz}V'_{gjz}A_c\right)_{I,J+1}^{n} - \left(\frac{\alpha}{1-\alpha}\frac{\rho_v\rho_l}{\rho_m}V'_{gjz}V'_{gjy}A_c\right)_{I,J}^{n}.\end{aligned}$$

2.4.4. Lateral momentum conservation equation

By applying volume integral and semi-implicit discretization to Eq.(2.19), the following equation can be formulated as:

$$\int_{CV}\left[\frac{\partial}{\partial t}\left(\rho_m u_m\right)+\nabla\cdot\left(\rho_m u_m \vec{u}_m\right)\right]dV = \int_{CV}\left[-\frac{\partial P}{\partial x}+\sum_k \nabla\cdot\left(\alpha_k \tau_k^{ij}\right)-\nabla\cdot\left(\frac{\alpha}{1-\alpha}\frac{\rho_g\rho_l}{\rho_m}V'_{gjx}\vec{V}'_{gj}\right)\right]dV. \tag{2.57}$$

As same as the axial momentum derivation did, the first order upwind scheme is applied to the convection term. A plausible assumption is used to derive the discretized form that the transverse momentum transfer is neglected in the lateral momentum equation[1]. It means that when x direction momentum is concerned in Cartesian coordinates, the x direction momentum transfer via the surface orthogonal to y-direction is neglected. This assumption may not be valid when the orthogonal momentum transfer is no more negligible such as a case with the directed crossflow by spacer grids[51]. However, it is plausible in many other cases due to small area of orthogonal surfaces and small magnitude of crossflows. The discretized convection term with the first order upwind scheme is given as:

$$\begin{aligned}
&\int_{CV}\nabla\cdot\left(\rho_m u_m \vec{u}_m\right)dV \\
&=\int_S\left(\left(\rho_m\vec{u}_m\right)^n u_m^{n+1}\right)\cdot d\vec{S} \\
&=\left[\left(\rho_m u_m\right)^n s_i \Delta z_J u_m^{n+1}\right]_{IB,J}-\left[\left(\rho_m u_m\right)^n s_i \Delta z_J u_m^{n+1}\right]_{IA,J}+\left[\left(\rho_m w_m\right)^n s_i l_i u_m^{n+1}\right]_{i,j} \\
&\quad -\left[\left(\rho_m w_m\right)^n s_i l_i u_m^{n+1}\right]_{i,j-1} \\
&=F_{x,e}^n u_{m,e}^{n+1}-F_{x,w}^n u_{m,w}^{n+1}+F_{x,up}^n u_{m,n}^{n+1}-F_{x,dw}^n u_{m,s}^{n+1} \\
&=\left(F_{x,e}^n+\max[-F_{x,e}^n,0]\right)u_{m,i,J}^{n+1}-\max[-F_{x,e}^n,0]u_{m,i+1,J}^{n+1}-\max[F_{x,w}^n,0]u_{m,i-1,J}^{n+1} \\
&\quad+\left(\max[F_{x,w}^n,0]-F_{x,w}^n\right)u_{m,i,J}^{n+1}+\left(F_{x,up}^n+\max[-F_{x,up}^n,0]\right)u_{m,i,J}^{n+1} \\
&\quad-\max[-F_{x,up}^n,0]u_{m,i,J+1}^{n+1}-\max[F_{x,dw}^n,0]u_{m,i,J-1}^{n+1}+\left(\max[F_{x,dw}^n,0]-F_{x,dw}^n\right)u_{m,i,J}^{n+1}.
\end{aligned} \tag{2.58}$$

Unlike the axial momentum equation, the viscous shear stress in the lateral momentum equation contains the form loss only. The discretized form of it becomes as follows:

$$\int_{CV}\nabla\cdot\boldsymbol{\tau}_m^{ij}dV=-\left[\frac{K_x}{2l_i}\rho_m u_m\left|u_m\right|\right]_{i,J}^n l_i s_i\Delta z_J=-\left[\frac{K_x}{2}\rho_m u_m\left|u_m\right|\right]_{i,J}^n s_i\Delta z_J. \tag{2.59}$$

In the end, the discretized subchannel form of the lateral momentum equation is give as:

$$a_{x,P}u_{m,i,J}^{n+1} - a_{x,W}u_{m,i-1,J}^{n+1} - a_{x,E}u_{m,i+1,J}^{n+1} - a_{x,DW}u_{m,i,J-1}^{n+1} - a_{x,UP}u_{m,i,J+1}^{n+1} = \left[P_{IA,J}^{n+1} - P_{IB,J}^{n+1}\right]s_i\Delta z_J + S_{x,i,J} \quad . \qquad (2.60)$$

where $a_{x,E} = \max\left[-F_{x,e}^{n}, 0\right]$, $a_{x,W} = \max\left[F_{x,w}^{n}, 0\right]$,

$$a_{x,UP} = \max\left[-F_{x,up}^{n}, 0\right],$$

$$a_{x,DW} = \max\left[F_{x,dw}^{n}, 0\right],$$

$$a_{x,P} = \sum_{nb} a_{x,nb} + \frac{(\rho_m)_{i,J}^{n}}{\Delta t}s_i l_i \Delta z_J$$

$$S_{x,i,J} = -\left[\frac{K_x}{2}\rho_m u_m |u_m|\right]_{i,J}^{n} s_i\Delta z_J - \sum_{nnb}[CM]_{x,nnb}^{n} + \frac{(\rho_m)_{i,J}^{n}(u_m)_{i,J}^{n}}{\Delta t}s_i l_i \Delta z_J$$

$$\sum_{nnb}[CM]_{x,nnb}^{n} = \left[\left(\frac{\alpha}{1-\alpha}\frac{\rho_v\rho_l}{\rho_m}V'_{gjx}V'_{gjx}\right)_{IB,J}^{n} - \left(\frac{\alpha}{1-\alpha}\frac{\rho_v\rho_l}{\rho_m}V'_{gjx}V'_{gjx}\right)_{IA,J}^{n}\right]s_i\Delta z_J + \left[\left(\frac{\alpha}{1-\alpha}\frac{\rho_v\rho_l}{\rho_m}V'_{gjx}V'_{gjz}\right)_{i,j}^{n} - \left(\frac{\alpha}{1-\alpha}\frac{\rho_v\rho_l}{\rho_m}V'_{gjx}V'_{gjz}\right)_{i,j-1}^{n}\right]s_i l_i \quad .$$

2.4.5. Mixture energy conservation equation

Let us apply volume integral and semi-implicit discretization to Eq.(2.21) to yield:

$$\int_{CV}\left[\frac{\partial}{\partial t}(\rho_m h_m) + \nabla\cdot(\rho_m h_m \vec{u}_m)\right]dV = \int_{CV}\left[-\nabla\cdot\vec{Q}_k''^{T} + q_w''' + q_v''' + \frac{\partial P}{\partial t} - \nabla\cdot\left(\frac{\alpha\rho_v\rho_l}{\rho_m}(h_v - h_l)\vec{V}'_{jg}\right)\right]dV \quad . \qquad (2.61)$$

The discretized subchannel form of the mixture energy conservation equation is given as:

$$\begin{aligned}
&\frac{\left(\rho_m h_m\right)_{I,J}^{n+1}-\left(\rho_m h_m\right)_{I,J}^{n}}{\Delta t}A_{c,I,J}\Delta z_J+\sum_{ic=1}^{nb}\left[\left(\rho_m h_m\right)^{\bar{n}}\left(\vec{u}_m^{\,n+1}\cdot\vec{s}\right)\Delta z_J\right]_{ic,J}\\
&\quad+\left[\left(\rho_m h_m\right)^{\bar{n}}\left(w_m\right)^{n+1}A_c\right]_{I,j}-\left[\left(\rho_m h_m\right)^{\bar{n}}\left(w_m\right)^{n+1}A_c\right]_{I,j-1}\\
&=\sum_{ic}s_{ic}\left(\frac{\varepsilon}{l}\right)_{ic}^{n}\left[\left(\rho_m h_m\right)_{\langle ic\rangle,J}^{n}-\left(\rho_m h_m\right)_{I,J}^{n}\right]\Delta z_J+\frac{P_{I,J}^{n+1}-P_{I,J}^{n}}{\Delta t}A_{c,I,J}\Delta z_J\\
&\quad+q_w''\xi_c\Delta z_J+q_v'''A_{c,I,J}\Delta z_J-\sum_{nnb}\left[CE\right]_{nnb}^{n}
\end{aligned}\tag{2.62}$$

where

$$\begin{aligned}
\sum_{nnb}\left[CE\right]_{nnb}^{n}&=\sum_{ic}\left[\frac{\alpha\rho_v\rho_l}{\rho_m}\left(h_v-h_l\right)\left(\vec{V}_{jg}'\cdot\vec{n}\right)\right]_{ic,J}^{n}s_{ic}\Delta z_J\\
&\quad+\left[\frac{\alpha\rho_v\rho_l}{\rho_m}\left(h_v-h_l\right)V_{jgz}'\right]_{I,j}^{n}A_{c,I,j}-\left[\frac{\alpha\rho_v\rho_l}{\rho_m}\left(h_v-h_l\right)V_{jgz}'\right]_{I,j-1}^{n}A_{c,I,j-1}
\end{aligned}.$$

2.5. Numerical Solution Method

2.5.1. Derivation of pressure-velocity correction with SIMPLE-like algorithm

There is a problem to solve the discretized momentum equations. The momentum equations have unknowns which are the implicit pressure term as well as the implicit velocities. A numerical scheme called Semi-Implicit Method for Pressure-Linked Equations (SIMPLE)[20] was proposed by Patankar in 1980 to handle pressure-velocity coupling. It is based on a guess-and-correct procedure on the staggered grid. Let us rewrite Eq.(2.56) and Eq.(2.60) as:

$$\begin{aligned}
a_{x,P}u_{m,i,J}^{n+1}&=\sum_{nb}a_{x,nb}u_{m,nb}^{n+1}+\left[P_{IA,J}^{n+1}-P_{IB,J}^{n+1}\right]s_i\Delta z_J+S_{x,i,J}\\
a_{z,P}w_{m,I,j}^{n+1}&=\sum_{nb}a_{z,nb}w_{m,nb}^{n+1}+\left[P_{I,J}^{n+1}-P_{I,J+1}^{n+1}\right]A_{c,I,j}+S_{z,I,j}
\end{aligned}.$$

For the beginning, the guessed pressure P^* is used instead of the pressure at the new time step (P^{n+1}). At the first iteration, the pressure at the previous time step (P^n) will be P^*. Afterward, the intermediate velocity u^* can be calculated as follows:

$$a_{x,P}u^{*}_{m,i,J} = \sum_{nb} a_{x,nb}u^{*}_{m,nb} + \left[P^{*}_{IA,J} - P^{*}_{IB,J} \right] s_i \Delta z_J + S_{x,i,J} \qquad (2.63)$$

$$a_{z,P}w^{*}_{m,I,j} = \sum_{nb} a_{z,nb}w^{*}_{m,nb} + \left[P^{*}_{I,J} - P^{*}_{I,J+1} \right] A_{c,I,j} + S_{z,I,j} \qquad (2.64)$$

Let us subtract the above Eq. (2.63) and Eq.(2.64) from the Eq. (2.56) and (2.60) respectively. Then the following relations between the velocity correction and pressure correction can be derived as:

$$a_{x,P}\left(u^{n+1}_{m,i,J} - u^{*}_{m,i,J}\right) = \sum_{nb} a_{x,nb}\left(u^{n+1}_{m,nb} - u^{*}_{m,nb}\right) + \left[\left(P^{n+1}_{IA,J} - P^{*}_{IA,J}\right) - \left(P^{n+1}_{IB,J} - P^{*}_{IB,J}\right)\right] s_i \Delta z_J \qquad (2.65)$$

$$a_{z,P}\left(w^{n+1}_{m,I,j} - w^{*}_{m,I,j}\right) = \sum_{nb} a_{z,nb}\left(w^{n+1}_{m,nb} - w^{*}_{m,nb}\right) + \left[\left(P^{n+1}_{I,J} - P^{*}_{I,J}\right) - \left(P^{n+1}_{I,J+1} - P^{*}_{I,J+1}\right)\right] A_{c,I,j} . \qquad (2.66)$$

In SIMPLE algorithm, the velocity corrections at neighboring cells are ignored, which is can be expressed as:

$$\sum_{nb} a_{x,nb}\left(u^{n+1}_{m,nb} - u^{*}_{m,nb}\right) \approx 0 \qquad (2.67)$$

$$\sum_{nb} a_{z,nb}\left(w^{n+1}_{m,nb} - w^{*}_{m,nb}\right) \approx 0 . \qquad (2.68)$$

Applying those assumptions to the Eq.(2.65) and Eq.(2.66), ones can obtain the final relations of velocity-pressure correction as below:

$$a_{x,P}\left(u^{n+1}_{m,i,J} - u^{*}_{m,i,J}\right) = \left[\left(P^{n+1}_{IA,J} - P^{*}_{IA,J}\right) - \left(P^{n+1}_{IB,J} - P^{*}_{IB,J}\right)\right] s_i \Delta z_J \qquad (2.69)$$

$$a_{z,P}\left(w^{n+1}_{m,I,j} - w^{*}_{m,I,j}\right) = \left[\left(P^{n+1}_{I,J} - P^{*}_{I,J}\right) - \left(P^{n+1}_{I,J+1} - P^{*}_{I,J+1}\right)\right] A_{c,I,j} \qquad (2.70)$$

However, the assumption that the velocity corrections at the neighboring cells are approximated to zero is rough. The under-relaxation scheme has to be applied to compensate the rough assumption. There are many variations of the SIMPLE Algorithm. One of the SIMPLE-like algorithm is SIMPLEC[52] (SIMPLE Consistent) proposed by Van Doormal and Raithby in 1984. Instead of the zero velocity correction at neighboring cells, they assumed that the velocity corrections at neighboring cells are regarded as same as the correction at diagonal term, which is expressed as below:

$$a_{x,nb}\left(u^{n+1}_{m,nb} - u^{*}_{m,nb}\right) \approx a_{x,P}\left(u^{n+1}_{m,i,J} - u^{*}_{m,i,J}\right) \qquad (2.71)$$

$$a_{z,nb}\left(w^{n+1}_{m,nb} - w^{*}_{m,nb}\right) \approx a_{x,P}\left(w^{n+1}_{m,i,J} - w^{*}_{m,i,J}\right) . \qquad (2.72)$$

This leads the change to the relations of the velocity-pressure correction as follows:

$$u_{m,i,J}^{n+1} = u_{m,i,J}^{*} + d_{i,J}\left[\left(P_{IA,J}^{n+1} - P_{IA,J}^{*}\right) - \left(P_{IB,J}^{n+1} - P_{IB,J}^{*}\right)\right] \tag{2.73}$$

where $d_{i,J} = \dfrac{s_i \Delta z_J}{a_{x,P} - \sum_{nb} a_{x,nb}}$

and

$$w_{m,I,j}^{n+1} = w_{m,I,j}^{*} + d_{I,j}\left[\left(P_{I,J}^{n+1} - P_{I,J}^{*}\right) - \left(P_{I,J+1}^{n+1} - P_{I,J+1}^{*}\right)\right] \tag{2.74}$$

where $d_{I,j} = \dfrac{A_{c,I,j}}{a_{z,PP} - \sum_{nb} a_{z,nb}}$.

The SIMPLEC method achieves the improvement that the solution can be obtained with very little or no pressure under-relaxation. In this reason, the SIMPLEC method is chosen as a main solution algorithm in the code.

2.5.2. Derivation of the pressure linear system coupled with the scalar equations

The remaining task is to derive the linear system which consists of the pressure correction terms ($\nabla P = P^{n+1} - P^{*}$) or the pressure at the new time step (P^{n+1}) as unknowns. The original SIMPLE method derives the pressure correction equation by coupling the continuity equation with the velocity-pressure correction relations. Among three scalar equations in the DFMs, the mixture mass continuity equation might be one of candidates. In order to recall the discretized mixture mass continuity, Eq.(2.46) is rewritten as:

$$\frac{\rho_{m,I,J}^{n+1} - \rho_{m,I,J}^{n}}{\Delta t} A_{c,I,J} \Delta z_J + \sum_{ic=1}^{nb}\left[\rho_m^{\overline{n}}\left(\vec{u}_m^{n+1} \cdot \vec{s}\right)_{ic} \Delta z_J\right]_{ic,J} + \left[\rho_m^{\overline{n}} w_m^{n+1} A_c\right]_{I,j} - \left[\rho_m^{\overline{n}} w_m^{n+1} A_c\right]_{I,j-1} = 0 .$$

The implicit velocity can be replaced with the Eq.(2.73) and Eq.(2.74), and one can build the equation having the intermediate velocities and the pressure correction terms. The problem occurs due to the density at the new time step that is still remaining as unknown. The simplest solution is to assume that a fluid in a system is an incompressible fluid[46]. The density of the incompressible fluid does not change

by time, which can be expressed as below:

$$\frac{\partial \rho}{\partial t} = 0 . \tag{2.75}$$

However, this approach is not valid to codes which aim to handle two-phase flow, because the density of mixture will vary significantly by time. Because the density is the function of pressure and enthalpy, the energy equation should be coupled to build the pressure correction equation by linearization as the ICE[21] method does. CTF uses all scalar equations to derive the pressure correction matrix. The older version of CUPID[54], which is a multidimensional two-phase flow code developed by KAERI, also constructs the pressure correction linear system using same approaches, even though a continuity-based semi-implicit scheme is applied in the newer version[55]. In the code, all scalar equations coupled pressure correction equation is used as an initial step.

(1) Linearization of mixture continuity equation

The primary variables are void fraction, liquid enthalpy, and pressure, so other secondary variables should be converted as functions with the primary variables. Those works might be possible through linearization as below:

$$\phi(t+\Delta t) = \phi(t) + \frac{\partial \phi(t)}{\partial t}\Delta t + \cdots \tag{2.76}$$

and then the mixture density can be linearized as follows:

$$\begin{aligned}
\frac{\partial \rho_m(\alpha, \rho_v, \rho_l)}{\partial t} &= \frac{\partial \alpha}{\partial t}\left(\frac{\partial \rho_m}{\partial \alpha}\right)^n + \frac{\partial \rho_v}{\partial t}\left(\frac{\partial \rho_m}{\partial \rho_v}\right)^n + \frac{\partial \rho_l}{\partial t}\left(\frac{\partial \rho_m}{\partial \rho_l}\right)^n \\
&= \frac{\partial \alpha}{\partial t}\left(\rho_v^n - \rho_l^n\right) + \frac{\partial \rho_{v,sat}(P)}{\partial t}\alpha^n + \frac{\partial \rho_l(P,h_l)}{\partial t}\left(1-\alpha^n\right) \\
&= \frac{\partial \alpha}{\partial t}\left(\rho_v^n - \rho_l^n\right) + \frac{\partial P}{\partial t}\left(\frac{\partial \rho_v}{\partial P}\right)^n \alpha^n + \left[\frac{\partial P}{\partial t}\left(\frac{\partial \rho_l}{\partial P}\right)^n + \frac{\partial h_l}{\partial t}\left(\frac{\partial \rho_l}{\partial h_l}\right)^n\right]\left(1-\alpha^n\right).
\end{aligned} \tag{2.77}$$

If the first order approximation is taken for temporal derivative, then the mixture density at the new time step can be approximated as follows:

$$\begin{aligned}
\rho_m^{n+1} = \rho_m^n &+ \left(\alpha^{n+1} - \alpha^n\right)\left(\rho_v^n - \rho_l^n\right) + \left(P^{n+1} - P^n\right)\left(\frac{\partial \rho_v}{\partial P}\right)^n \alpha^n \\
&+ \left[\left(P^{n+1} - P^n\right)\left(\frac{\partial \rho_l}{\partial P}\right)^n + \left(h_l^{n+1} - h_l^n\right)\left(\frac{\partial \rho_l}{\partial h_l}\right)^n\right]\left(1-\alpha^n\right).
\end{aligned} \tag{2.78}$$

By Inserting the Eq.(2.78) into the Eq.(2.46) and rearranging unknowns to LHS and source terms and $\vec{u}_m^{n+1}$ to RHS, the linearized mixture continuity equation can be derived as follows:

$$\begin{aligned} & a_{11}h_{l,I,J}^{n+1} + a_{12}\alpha_{I,J}^{n+1} + a_{13}P_{I,J}^{n+1} \\ & = -\left[\sum_{ic=1}^{nb}\left[\rho_m^{\bar{n}}\left(\vec{u}_m^{n+1}\cdot\vec{s}\right)_{ic}\Delta z_J \right]_{ic,J} + \left[\rho_m^{\bar{n}} w_m^{n+1} A_c \right]_{I,j} - \left[\rho_m^{\bar{n}} w_m^{n+1} A_c \right]_{I,j-1} \right] + s_1 \end{aligned} \tag{2.79}$$

where $a_{11} = \left(\frac{\partial \rho_l}{\partial h_l}\right)_{I,J}^n \left(1-\alpha_{I,J}^n\right)\left(\frac{A_{c,I,J}\Delta z_J}{\Delta t}\right)$

$$a_{12} = \left(\rho_{v,I,J}^n - \rho_{l,I,J}^n\right)\left(\frac{A_{c,I,J}\Delta z_K}{\Delta t}\right)$$

$$a_{13} = \left(\left(\frac{\partial \rho_v}{\partial P}\right)_{I,J}^n \alpha_{I,J}^n + \left(\frac{\partial \rho_l}{\partial P}\right)_{I,J}^n \left(1-\alpha_{I,J}^n\right)\right)\left(\frac{A_{c,I,J}\Delta z_J}{\Delta t}\right)$$

$$s_1 = a_{11}h_{l,I,J}^n + a_{12}\alpha_{I,J}^n + a_{13}P_{I,J}^n \,.$$

(2) Linearization of vapor continuity equation

In the vapor continuity equation, the vapor density and the vapor/liquid temperature are necessary to be linearized as:

$$\rho_v^{n+1} = \rho_v^n + \left(P^{n+1} - P^n\right)\left(\frac{\partial \rho_v}{\partial P}\right)^n \tag{2.80}$$

$$T_l^{n+1} = T_l^n + \left(P^{n+1} - P^n\right)\left(\frac{\partial T_l}{\partial P}\right)^n + \left(h_l^{n+1} - h_l^n\right)\left(\frac{\partial T_l}{\partial h_l}\right)^n \tag{2.81}$$

$$T_{v,sat}^{n+1} = T_{v,sat}^n + \left(P^{n+1} - P^n\right)\left(\frac{\partial T_{v,sat}}{\partial P}\right)^n . \tag{2.82}$$

Inserting the above relations into Eq.(2.49) and rearranging terms can derive the linearized vapor continuity equation as follows:

$$\begin{aligned} & a_{21}h_{I,J}^{n+1} + a_{22}\alpha_{I,J}^{n+1} + a_{23}P_{I,J}^{n+1} \\ & = -\left[\begin{array}{l} \sum_{ic=1}^{nb}\left[\left(\alpha\rho_v\right)^{\bar{n}}\left(\vec{u}_m^{n+1}\cdot\vec{s}\right)\Delta z_J \right]_{ic,J} + \\ \left[\left(\alpha\rho_v\right)^{\bar{n}} w_m^{n+1} A_c \right]_{I,j} - \left[\left(\alpha\rho_v\right)^{\bar{n}} w_m^{n+1} A_c \right]_{I,j-1} \end{array} \right] + s_2 \end{aligned} \tag{2.83}$$

where $a_{21} = -\dfrac{H_{il}^n}{h_{v,sat}^n - h_{il}^n}\left(\dfrac{\partial T_l}{\partial h_l}\right)_{I,J}^n A_{c,I,J}\Delta z_J$

$$a_{22} = \rho_{v,I,J}^n\left(\frac{A_{c,I,J}\Delta z_J}{\Delta t}\right)$$

$$a_{23} = \alpha_{I,J}^n\left(\frac{\partial \rho_v}{\partial P}\right)_{I,J}^n\left(\frac{A_{c,I,J}\Delta z_J}{\Delta t}\right) + \frac{H_{il}^n}{h_{v,sat}^n - h_{il}^n}\left[\left(\frac{\partial T_{v,sat}}{\partial P}\right)_{I,J}^n - \left(\frac{\partial T_l}{\partial P}\right)_{I,J}^n\right]A_{c,I,J}\Delta z_J$$

$$s_2 = \left(\frac{H_{il}^n(T_l^n - T_{v,sat}^n)}{h_{v,sat}^n - h_{li}^n}\right)_{I,J} A_{c,I,J}\Delta z_J - \sum_{nnb}[CV]_{nnb}^n + a_{21}h_{l,I,J}^n + a_{22}\alpha_{I,J}^n + a_{23}P_{I,J}^n .$$

(3) Linearization of mixture energy equation

The linearization of mixture density can be formulated as:

$$\begin{aligned}(\rho_m h_m)^{n+1} &= (\rho_m h_m)^n + \delta\alpha\left((\rho_v h_v)^n - (\rho_l h_l)^n\right) \\ &+\left(\delta P\left(\frac{\partial \rho_l}{\partial P}\right)^n + \delta h_l\left(\frac{\partial \rho_l}{\partial h_l}\right)^n\right)(1-\alpha^n)h_l^n + \delta P\left(\frac{\partial \rho_v}{\partial P}\right)^n(\alpha h_v)^n \\ &+\delta h_l(1-\alpha^n)\rho_l^n + \delta P\left(\frac{\partial h_v}{\partial P}\right)^n(\alpha\rho_v)^n\end{aligned} \quad (2.84)$$

where $\delta\phi = \phi^{n+1} - \phi^n$.

By replacing the corresponding terms of Eq. (2.62) with that of Eq.(2.84) and rearranging unknowns and knowns, the following linearized form of mixture energy equation can be formulated as follows:

$$\begin{aligned}&a_{31}h_{l,I,J}^{n+1} + a_{32}\alpha_{I,J}^{n+1} + a_{33}P_{I,J}^{n+1} \\ &= -\left[\begin{array}{l}\displaystyle\sum_{ic=1}^{nb}\left[(\rho_m h_m)^{\overline{n}}(\vec{u}_m^{n+1}\cdot\vec{s})\Delta z_J\right]_{ic,J} + \left[(\rho_m h_m)^{\overline{n}}(w_m)^{n+1}A_c\right]_{I,j} \\ -\left[(\rho_m h_m)^{\overline{n}}(w_m)^{n+1}A_c\right]_{I,j-1}\end{array}\right] + s_3 .\end{aligned} \quad (2.85)$$

where

$$a_{31} = \left(1-\alpha_{I,J}^{n}\right)\left(\left(\frac{\partial \rho_l}{\partial h_l}\right)_{I,J}^{n} h_{l,I,J}^{n} + \rho_{l,I,J}^{n}\right)\frac{A_{c,I,J}\Delta z_J}{\Delta t}$$

$$a_{32} = \left(\left(\rho_v h_v\right)_{I,J}^{n} - \left(\rho_l h_l\right)_{I,J}^{n}\right)\frac{A_{c,I,J}\Delta z_J}{\Delta t}$$

$$a_{33} = \left(h_{l,I,J}^{n}\left(1-\alpha_{I,J}^{n}\right)\left(\frac{\partial \rho_l}{\partial P}\right)_{I,J}^{n} + \left(\frac{\partial \rho_v}{\partial P}\right)_{I,J}^{n}\left(\alpha h_v\right)_{I,J}^{n} + \left(\frac{\partial h_v}{\partial P}\right)_{I,J}^{n}\left(\alpha \rho_v\right)_{I,J}^{n} - 1\right)\left(\frac{A_{c,I,J}\Delta z_J}{\Delta t}\right)$$

$$s_3 = \sum_{ic} s_{ic}\left(\frac{\varepsilon}{l}\right)_{ic}^{n}\left[\left(\rho_m h_m\right)_{\langle ic\rangle,J}^{n} - \left(\rho_m h_m\right)_{I,J}^{n}\right]\Delta z_J + q_w'' \xi_c \Delta z_J + q_v''' A_{c,I,J}\Delta z_J$$
$$- \sum_{nnb}\left[CE\right]_{nnb}^{n} + a_{31} h_{l,I,J}^{n} + a_{32}\alpha_{I,J}^{n} + a_{33} P_{I,J}^{n}$$

(4) SIMPLE algorithm with the linear system involving the pressure equation

In Cartesian coordinates, a linear system can be constructed at a certain scalar cell (*I,J*) by using three linearized scalar equations as below:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}\begin{bmatrix} h_{l,I,J}^{n+1} \\ \alpha_{I,J}^{n+1} \\ P_{I,J}^{n+1} \end{bmatrix} = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix} - \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} & b_{15} & b_{16} \\ b_{21} & b_{22} & b_{23} & b_{24} & b_{25} & b_{26} \\ b_{31} & b_{32} & b_{33} & b_{23} & b_{35} & b_{36} \end{bmatrix}\begin{bmatrix} u_{ie,J}^{n+1} \\ -u_{iw,J}^{n+1} \\ v_{is,J}^{n+1} \\ -v_{in,J}^{n+1} \\ w_{I,j}^{n+1} \\ -w_{I,j-1}^{n+1} \end{bmatrix}. \qquad (2.86)$$

Let us replace $\vec{u}^{n+1}$ with the relations of the pressure-velocity correction (Eq.(2.73), Eq.(2.74)) and take the inverse of the 3x3 coefficient matrix of LHS at Eq.(2.86). Then one can derive a following linear system as:

$$\begin{bmatrix} h_{l,I,J}^{n+1} \\ \alpha_{I,J}^{n+1} \\ P_{I,J}^{n+1} \end{bmatrix} = \begin{bmatrix} s_1' \\ s_2' \\ s_3' \end{bmatrix} - \begin{bmatrix} b_{11}' & b_{12}' & b_{13}' & b_{14}' & b_{15}' & b_{16}' \\ b_{21}' & b_{22}' & b_{23}' & b_{24}' & b_{25}' & b_{26}' \\ b_{31}' & b_{32}' & b_{33}' & b_{23}' & b_{35}' & b_{36}' \end{bmatrix} \begin{bmatrix} d_{ie,J}\left[P_{I,J}^{n+1} - P_{IE,J}^{n+1}\right] \\ d_{iw,J}\left[P_{I,J}^{n+1} - P_{IW,J}^{n+1}\right] \\ d_{is,J}\left[P_{I,J}^{n+1} - P_{IS,J}^{n+1}\right] \\ d_{in,J}\left[P_{I,J}^{n+1} - P_{IN,J}^{n+1}\right] \\ d_{I,j}\left[P_{I,J}^{n+1} - P_{I,J+1}^{n+1}\right] \\ d_{I,j-1}\left[P_{I,J}^{n+1} - P_{I,J-1}^{n+1}\right] \end{bmatrix} \tag{2.87}$$

where

$$\begin{bmatrix} b_{11}' & b_{12}' & b_{13}' & b_{14}' & b_{15}' & b_{16}' \\ b_{21}' & b_{22}' & b_{23}' & b_{24}' & b_{25}' & b_{26}' \\ b_{31}' & b_{32}' & b_{33}' & b_{23}' & b_{35}' & b_{36}' \end{bmatrix} = \left(\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \right)^{-1} \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} & b_{15} & b_{16} \\ b_{21} & b_{22} & b_{23} & b_{24} & b_{25} & b_{26} \\ b_{31} & b_{32} & b_{33} & b_{23} & b_{35} & b_{36} \end{bmatrix}$$

and

$$\begin{bmatrix} s_1' \\ s_2' \\ s_3' \end{bmatrix} = \left(\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \right)^{-1} \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix} - \begin{bmatrix} b_{11}' & b_{12}' & b_{13}' & b_{14}' & b_{15}' & b_{16}' \\ b_{21}' & b_{22}' & b_{23}' & b_{24}' & b_{25}' & b_{26}' \\ b_{31}' & b_{32}' & b_{33}' & b_{23}' & b_{35}' & b_{36}' \end{bmatrix} \left(\begin{bmatrix} u_{ie,J}^* \\ -u_{iw,J}^* \\ v_{is,J}^* \\ -v_{in,J}^* \\ w_{I,j}^* \\ -w_{I,j-1}^* \end{bmatrix} + \begin{bmatrix} d_{ie,J}\left[-P_{I,J}^* + P_{IE,J}^*\right] \\ d_{iw,J}\left[-P_{I,J}^* + P_{IW,J}^*\right] \\ d_{is,J}\left[-P_{I,J}^* + P_{IS,J}^*\right] \\ d_{in,J}\left[-P_{I,J}^* + P_{IN,J}^*\right] \\ d_{I,j}\left[-P_{I,J}^* + P_{I,J+1}^*\right] \\ d_{I,j-1}\left[-P_{I,J}^* + P_{I,J-1}^*\right] \end{bmatrix} \right).$$

The above linear system can be established at each scalar cell. The solution of the linear system is decoupled with neighboring cells except the pressure. By extracting elements in the 3rd row, the pressure equation is derived as follows:

$$-b_{36}' P_{I,J-1}^{n+1} - b_{34}' P_{IN,J}^{n+1} - b_{32}' P_{IW,J}^{n+1} + \left[1 + \sum_{nb} b_{nb}'\right] P_{I,J}^{n+1} - b_{31}' P_{IE,J}^{n+1} - b_{33}' P_{IS,J}^{n+1} - b_{35}' P_{I,J+1}^{n+1} = s_3' . \tag{2.88}$$

A linear system can be derived as shown at Figure 2-5, which has solutions by pressures at the new time step and septa-diagonal matrix.

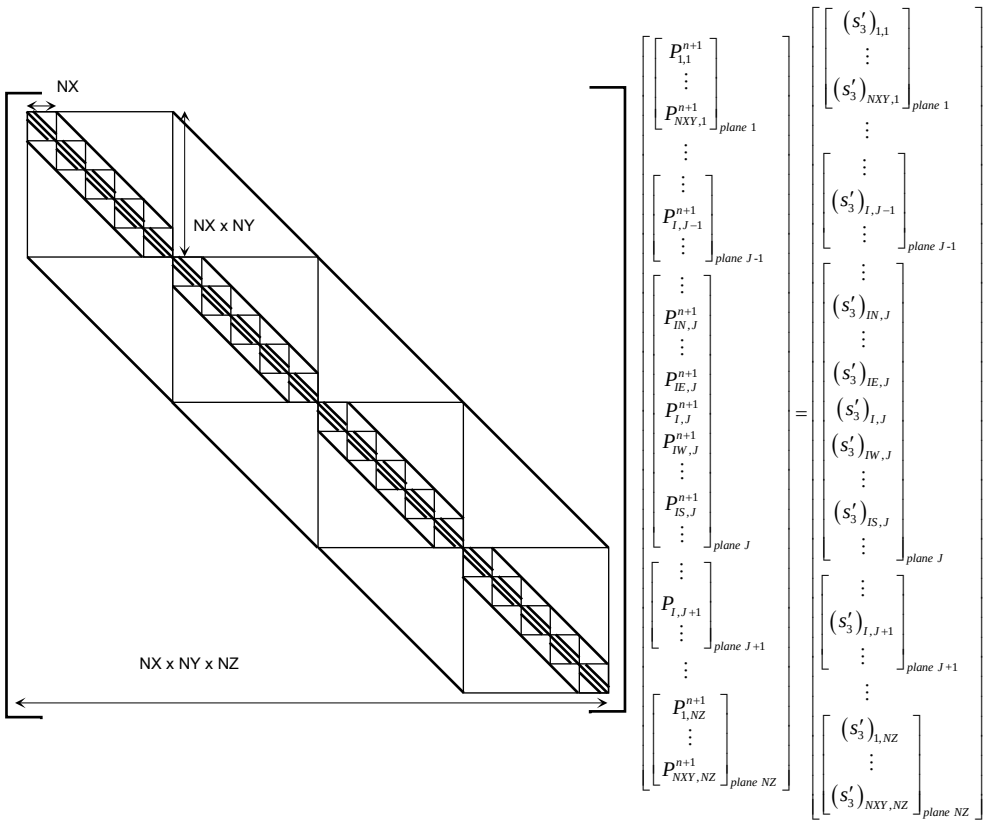

Figure 2-5. Linear system involving a septa-diagonal pressure matrix

The velocities at the new time step will be updated by Eq.(2.73) and Eq.(2.74) with newly obtained pressures. Because the updated velocities at the next time step satisfy the pressure correction relation but not the momentum conservation, the iteration process might be necessary until a certain criterion. This iteration is called as outer iteration. The convergence criterion is frequently given as:

$$\frac{\left\| P^{n+1} - P^{*} \right\|}{\left\| P^{n+1} \right\|} \leq \varepsilon \,. \tag{2.89}$$

After the converged velocities and pressures at the next step are obtained, the remaining scalar variables will be updated by Eq.(2.87). The entire process of the SIMPLEC algorithm is shown in Figure 2-6 in step by step.

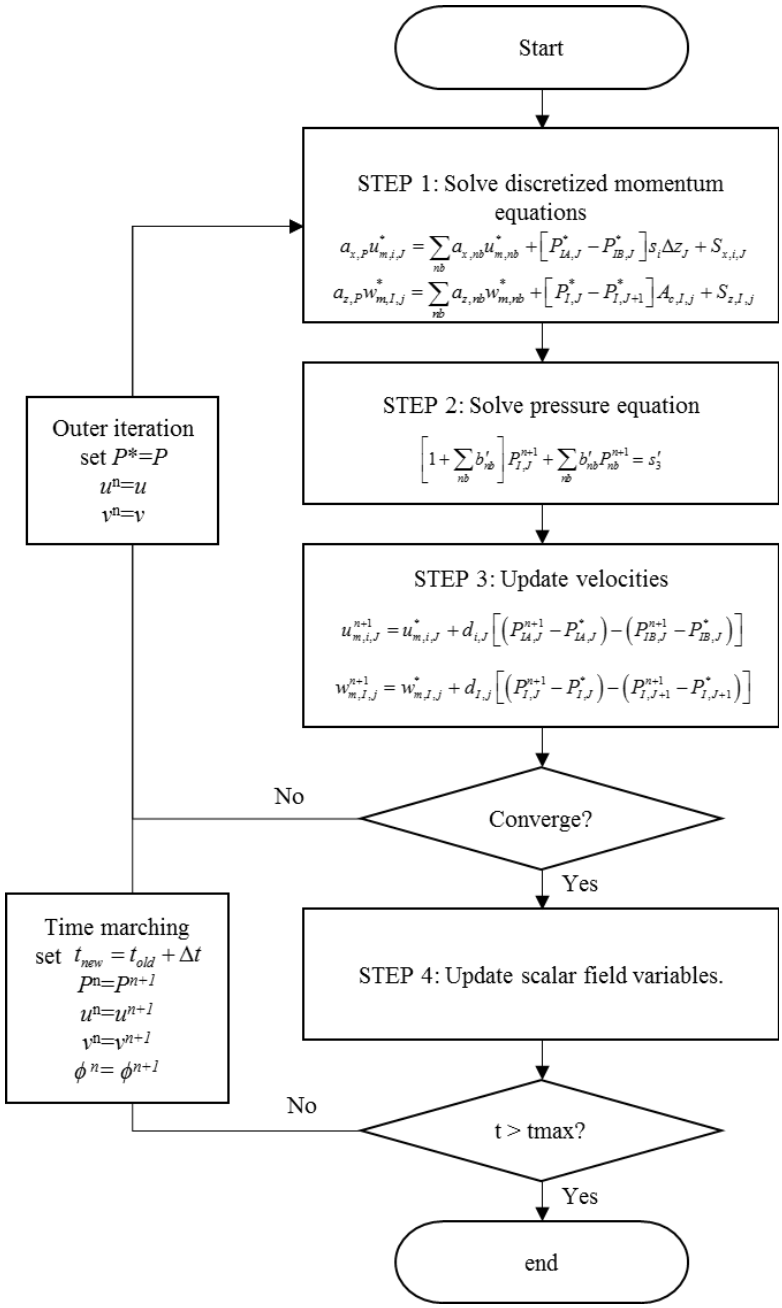

Figure 2-6. SIMPLEC Algorithm

Chapter 3. Development and Validation of the α-version Code

3.1. Description of the developed code

3.1.1. Introduction of ESCOT

The Efficient Simulator of COre Thermal-hydraulics (ESCOT) code is developed in α-version. The code is written in the standard Fortran 2003 language. The code is designed to perform the T/H analysis of two-phase flow in the subchannel level with the rectangular geometry as PWRs or BWRs. The drift-flux model is applied to analyze two-phase flow and the SIMPLEC algorithm is used for the steady-state and transient solver. The preprocessor is also developed to easily provide a subchannel-level input to users with simple core geometry information, initial and boundary condition.

3.1.2. Code structure

The most outer structure of ESCOT is simple as shown as Figure 3-1. When the code is executed, a subroutine ‘init’ is firstly called. The function of ‘init’ subroutine is to read an input deck and allocate/initialize all variables required for a calculation.

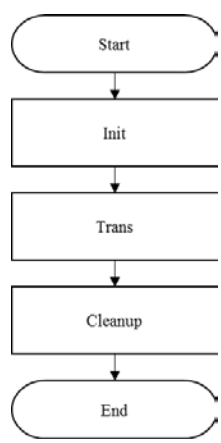

Figure 3-1. The structure of ESCOT

The main calculation is performed by a subroutine ‘Trans’. The structure of ‘Trans’ is illustrated in Figure 3-2. The structure of ‘Trans’ is almost same as the algorithm of SIMPLC. Although the outer iteration loop is implemented in ESCOT to obtain the converged pressures and velocities at the next time step, a one-through-scheme which does not perform outer iteration is preferable in the current version. Instead, a short time step size (or small Courant number) is kept during a calculation to compensate the absence of outer iteration. The effect of outer iteration and under-relaxation for long time step calculations will be tested in the future. After a transient calculation is done, the code finally calls a subroutine ‘clean up’ that deallocates memory and prints outputs before the code is terminated.

3.1.3. Pseudo steady-state

ESCOT is a transient code and it does not have an explicit steady-state solver based on governing equations where all temporal derivative terms are removed. ESCOT, instead, obtains steady-state solutions with null-transient calculations. Four convergence criteria are monitored to determine whether a calculation reaches steady-state.

The first parameter is the mass balance. It checks the outlet mass is equal to the inlet mass. It is defined as the difference between inlet and outlet mass flow rate divided by inlet mass flow rate as below:

$$M_{\text{balance}}(\%) = \frac{\dot{m}_{inlet} - \dot{m}_{outlet}}{\dot{m}_{inlet}} \cdot 100 \quad \text{where} \quad \dot{m}_{in} = \sum_{cell,in} \left(\rho v_m A\right)_{cell} . \qquad (3.1)$$

The second parameter is the energy balance. It represents the energy conservation by checking the energy coming out via outlet, energy coming in via inlet and via heated rod as follows:

$$E_{balance}(\%) = \frac{Q_{outlet} - \left(Q_{rod} + Q_{inlet}\right)}{Q_{rod} + Q_{inlet}} \cdot 100 \quad \text{where} \quad Q_{in} = \sum_{cell,in} \left(\dot{m} h_m\right)_{cell} . \qquad (3.2)$$

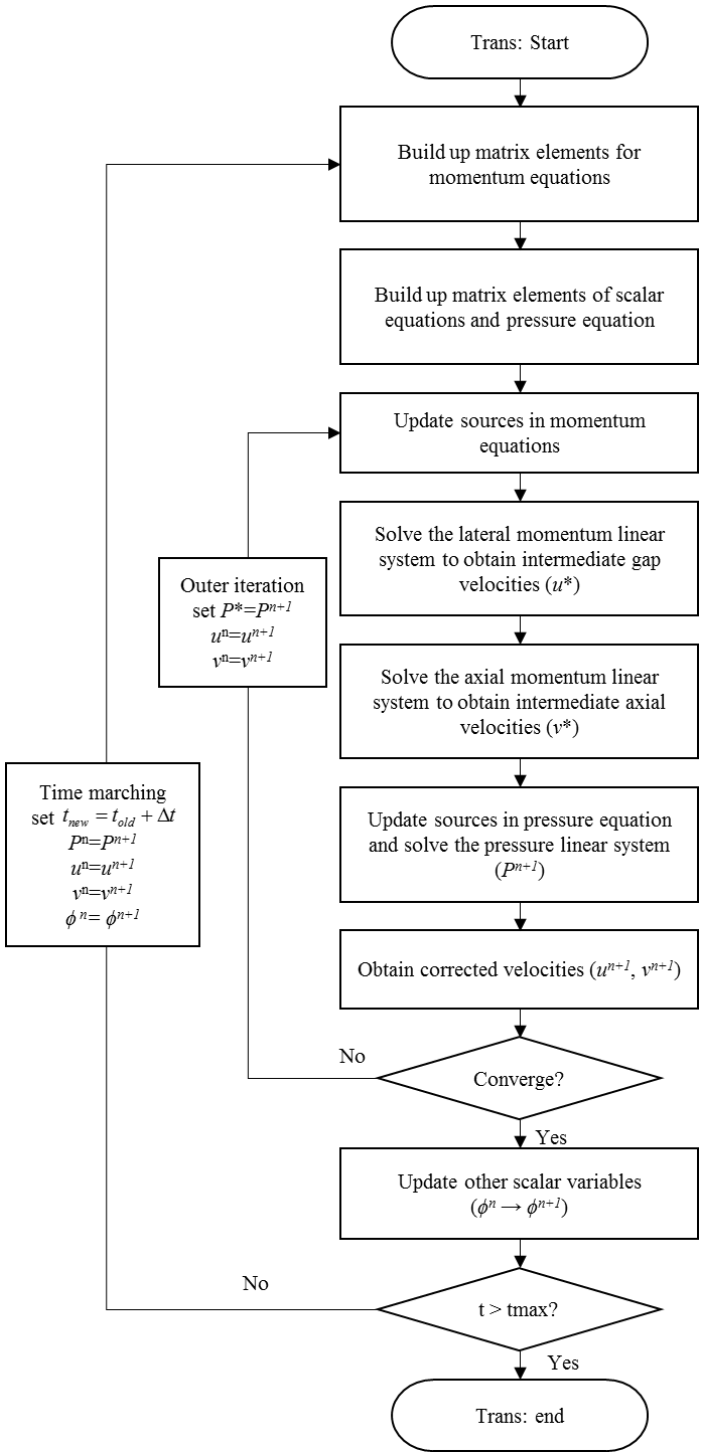

Figure 3-2. The structure of Trans

The third parameter is the mass storage. It accounts for mass stored in the system during the time step. All cells are taken into account by comparing the difference between the new time step value and the old time step value to calculate the mass storage. It is expressed as:

$$\Delta M_{\text{mass storage}}(\%) = \frac{\Delta M_{fluid}}{\dot{m}_{inlet}} \cdot 100 \tag{3.3}$$

where $\Delta M_{fluid} = \sqrt{\sum_{cell}^{nell} \left[\left(\rho_{m,cell}^{n+1} - \rho_{m,cell}^{n} \right) \Delta V_{cell} \right]^2} \; / \; \Delta t$.

The final parameter is the fluid energy storage. It accounts for energy stored in the fluid during the time step. The same approach with the mass storage is used, and the expression is as follows:

$$\Delta E_{\text{fluid storage}} = \frac{\Delta E_{fluid}}{Q_{rod} + Q_{fluid}} \cdot 100 \tag{3.4}$$

where $\Delta E_{fluid} = \sqrt{\sum_{cell}^{nell} \left[\left(\left(\rho_m h_m \right)_{cell}^{n+1} - \left(\rho_m h_m \right)_{cell}^{n} \right) \Delta V_{cell} \right]^2} \; / \; \Delta t$.

3.2. Verification

3.2.1. Pressure drop by gravity

This verification, one of the simplest tests, has been performed to check that the gravitational force is properly implemented in the code. From the axial momentum equation, it is assumed that:

1) The case is steady state which makes eliminate the temporal derivative term.
2) Do not consider external forces except pressure and gravity force.
3) The crossflow is negligible and the axial velocity distribution and densities are constant over the problem domian. Thus, the convection term can be removed.

Therefore, the pressure gradient is affected by only the gravitational acceleration.

$$\frac{\partial P}{\partial z} = -\rho g \tag{3.5}$$

The test is performed with a 7x7 channel bundle geometry. The distance of center-

to-center of channel is 1.2cm and each channel has uniform inlet mass flux 2,967.1kg/m^2. The gap thickness is set to 0.003m. The height of problem is 10.16m which is sufficiently high to observe the pressure drop by the gravity. Axial meshing is set to 25.4mm (1 inch), so the number of total meshes is 19,600 (7×7×400). The inlet temperature is set to 200^{o}C to keep sufficiently subcooled and the outlet pressure is 15.513 MPa. The constant density, 894.9kg/m^3, is used. The case is unheated.

Figure 3-3 demonstrates excellent agreement between the results from the code and analytic solution. The maximum relative error is 8.45×10^{-6}% and the RMS of errors is 3.87×10^{-6}%. In conclusion, the gravitational force in the axial momentum equation is implemented well in the code.

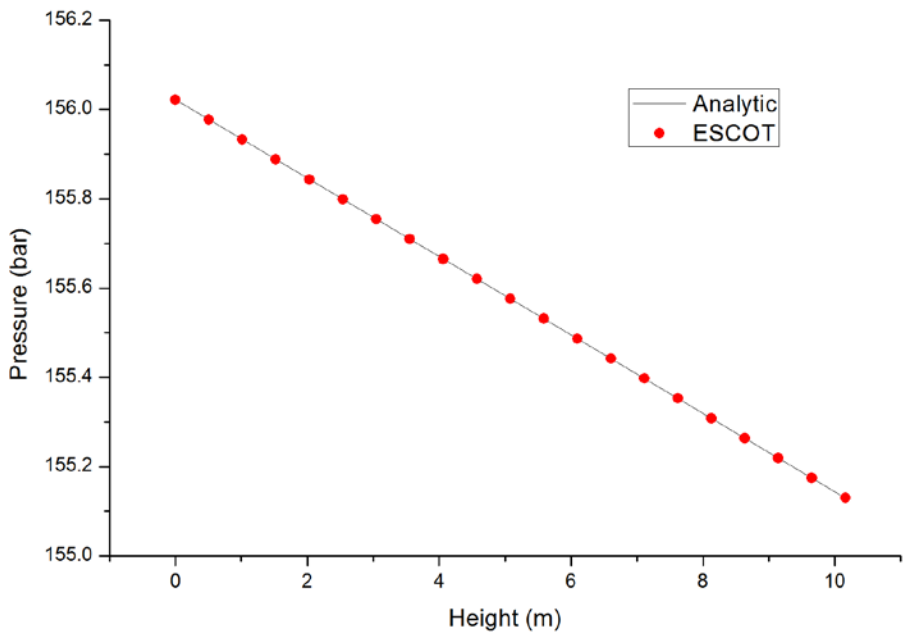

Figure 3-3. Comparison bundle pressure from ESCOT with analytic solution

3.2.2. Single-phase Friction Flow Split

(1) Problem background

There are two driving forces for crossflow in ESCOT: pressure difference, and turbulent mixing. If the turbulent mixing model is disabled, then the pressure difference is the only driving force of crossflow. The lateral pressure gradient might arise due to lateral density difference by non-uniform radial heating, non-uniform inlet conditions, or different pressure drop in adjacent channels. In this verification, the crossflow driven by different pressure drop will be tested with an unheated and uniform inlet boundary case, as CTF did[56].

Three pressure drop models are implemented in ESCOT: gravitational acceleration, friction loss and form loss. In case of the gravitational force, it uniformly acts in the whole body, it does not yield different pressure drops between channels. On the other hands, by assigning different hydraulic diameters, different pressure drops in channels are easily simulated. Therefore, the friction loss will be governing force in this verification.

In addition, the crossflow accompanies the flow redistribution phenomenon. The flow redistribution mechanism can be explained qualitatively as follows. Let us consider two adjacent channels. If one channel has a higher pressure drop than other, the lateral pressure gradient will be created from the higher resistance channel to the lower one. The lateral pressure gradient drives a crossflow with the same direction as the lateral pressure gradient. Thus, the flow starts to be redistributed, as the velocity of the low-resistance channel increases and, meanwhile, that of the high-resistance channel decreases. However, the increased velocity in the low-resistance channel induces a higher friction loss than before, whereas the opposite phenomenon happens in the high-resistance channel. The flow redistribution will be ended, when both channel reach the equilibrium state where the pressure drops in two channels are equal.

In this test, three factors will be verified.

1) Flow redistribution phenomenon by crossflow that occurs by different pressure drop in adjacent channels.
2) Mass flow rate when it reaches fully-redistributed state. The analytic solution will be given.
3) Friction loss model in the code

(2) Derivation of analytic solution

In order to drive an analytic solution, a case consisting of two channels is considered. In addition, it is assumed that:

1) The case is steady state which makes eliminate the temporal derivative term.
2) The crossflow is negligible, so the lateral convection can be eliminated.
3) The axial velocity distribution as well as the density in the control volume is constant. Thus, the convection term can be removed.

Therefore, the axial momentum equation can be simplified as follows:

$$\frac{\partial P}{\partial z} = -\rho g + \frac{\partial \tau}{\partial z}. \tag{3.6}$$

At equilibrium state, the pressure drops as well as the gravitational acceleration are same. Therefore, the following relation can be derived by integrating over z in the control volume as:

$$\tau_{w,1} = \tau_{w,2}. \tag{3.7}$$

Because the wall friction model is formulated by Eq.(2.29), Eq.(3.7) is equal to:

$$\frac{f_1 w_1^2}{2D_{h,1}} = \frac{f_2 w_2^2}{2D_{h,2}}. \tag{3.8}$$

For the friction factor, the McAdams correlation (Table 2-1) is used. By Substituting the friction factor in the Eq.(3.8) with the McAdams correlation and doing some arithmetic, the one can derive below relation as:

$$\left(\frac{w_1}{w_2}\right)^{2+C_2} = \left(\frac{D_{h,2}}{D_{h,1}}\right)^{C_2-1} \tag{3.9}$$

where $C_2 = -0.2$ in the McAdams correlation.

The velocity terms in the above relation can be replaces with the mass flow rate terms as:

$$\left(\frac{\dot{m}_1 A_2}{\dot{m}_2 A_1}\right)^{2+C_2} = \left(\frac{D_{h,2}}{D_{h,1}}\right)^{C_2-1}. \tag{3.10}$$

Because the inlet mass flow rate satisfies as:

$$\dot{m}_{in} = \dot{m}_1 + \dot{m}_2, \tag{3.11}$$

the analytic solutions of mass flow rates at the fully-redistributed point become:

$$\dot{m}_2 = \frac{\dot{m}_{in}}{\left(1+\left(\frac{D_{h,2}}{D_{h,1}}\right)^{\frac{C_2-1}{2+C_2}}\frac{A_1}{A_2}\right)} \tag{3.12}$$

and

$$\dot{m}_1 = \dot{m}_{in} - \dot{m}_2 \,. \tag{3.13}$$

(3) Test problem description

A test problem is set to consist of two channels with different hydraulic diameters. The detail descriptions of geometrical information inlet and outlet boundary conditions of the test problem are illustrated at Figure 3-4. The case is unheated. Two channels are linked with a gap whose thickness is 3.00mm and length is 12.6mm. The axial height is 9.996m which is enough high to observe the split flow at equilibrium state. The axial meshing is set to 0.102m which is 98 planes in total. The expected mass flow at the equilibrium state is 0.2210kg/sec in channel 1 and 0.7017kg/sec in channel 2.

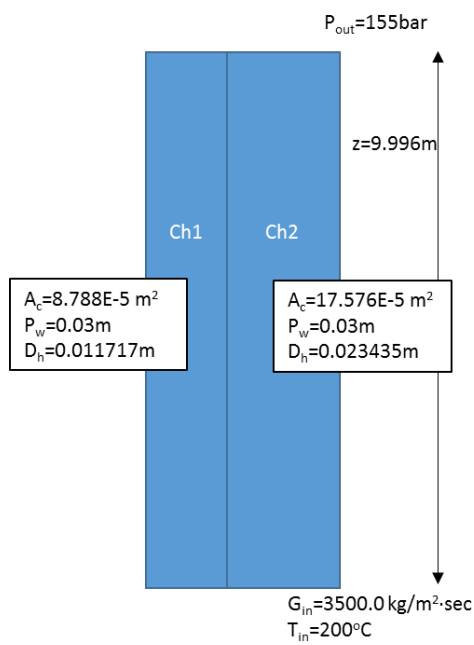

Figure 3-4. Geometry and inlet boundary condition of friction flow split test

(4) Results

The calculated results are converted to the normalized mass flux by the inlet mass flux as defined as:

$$G_{i,norm}(\%) = \frac{G_i - G_{in}}{G_{in}} \times 100 . \quad (3.14)$$

The analytic solutions as the normalized mass flux are -28.14% for channel 1 and 14.07% for channel 2 in this problem. The computed results of ESCOT and CTF are shown at Figure 3-5.

Firstly, the flow redistribution phenomenon is observed from calculated results. The mass flux of ch. 2, which has low-resistance, is increased at the beginning and it asymptotically saturated, while the opposite behavior is predicted in ch. 1. Secondly, the calculated normalized mass flux values at the Equilibrium state are -28.12% (ch. 1) and 14.08% (ch. 2), which is strongly agreed with the analytic solutions having -0.062% and 0.078% relative errors at ch. 1 and ch. 2 respectively. Thus, it can be concluded that the friction loss term is implemented well and ESCOT can analyze the crossflow by the different pressure loss between adjacent cells.

One point is here that the analytic solutions do not provide the flow redistribution rate, but only the value at the equilibrium state. Thus, the flow split rate is verified by the code-to-code comparison. As the results of CTF are set as the reference, the results of ESCOT have the maximum relative errors by 0.65% and 0.56%, and the RMS of errors by 0.21% and 0.20% at ch. 1 and ch. 2 respectively. Therefore, it can be concluded that the calculated flow split rates are also comparably accurate.

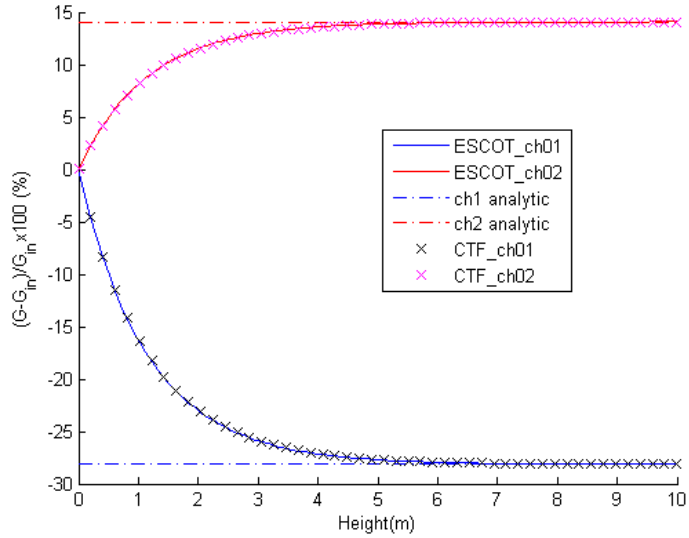

Figure 3-5. Relative mass flux along the height computed by ESCOT and analytic solution of Equilibrium mass flux in friction flow split test problem

3.2.3. Single-phase Two-Channel Turbulent Mixing

(1) Derivation of analytic solution

This verification is performed to check the energy exchange in adjacent channels by turbulent mixing model under the single-phase flow condition as CTF did[56]. The derivation of the analytic solution starts from the general liquid enthalpy equation by applying some assumptions as follows:

1) The case is steady state which makes eliminate the temporal derivative term.
2) The crossflow is negligible, so the lateral convection can be eliminated.
3) The axial velocity distribution as well as the density in the control volume is constant.
4) The case is unheated.

By the above assumptions, the general liquid enthalpy equation can be simplified as follows:

$$\frac{\partial \rho_l h_l w_l}{\partial z} = \sum_{IC} \frac{s_{I,IC}}{A_I} \left(\frac{\varepsilon}{l} \right)_{I,IC} \left[\left(\rho_l h_l \right)_{IC} - \left(\rho_l h_l \right)_I \right]. \tag{3.15}$$

The 3rd assumption makes possible to take out the density and axial velocity from the partial derivative of z. Then let us replace turbulent mixing coefficient with Eq.(2.36) which contains mixing parameter β. This leads Eq.(3.15) to be turned into:

$$\dot{m} \frac{\partial h_l}{\partial z} = \sum_{IC} s_{I,IC} \beta G \left[\left(h_l \right)_{IC} - \left(h_l \right)_I \right]. \tag{3.16}$$

If the case consisting of two channels is concerned, Eq.(3.16) of each channel can be expressed as follows:

$$\dot{m}_1 \frac{\partial h_1}{\partial z} + W'_{1\to 2} \left(h_1 - h_2 \right) = 0 \quad \text{(for ch. 01)} \tag{3.17}$$

$$\dot{m}_2 \frac{\partial h_2}{\partial z} - W'_{1\to 2} \left(h_1 - h_2 \right) = 0 \quad \text{(for ch. 02)} \tag{3.18}$$

where $W'_{1\to 2} = \beta s_{1,2} G$.

By the 3rd assumption, the relation $\dot{m}_1 = \dot{m}_2$ can be established, and the enthalpy balance equation written as:

$$\dot{m}_1 h_1 + \dot{m}_2 h_2 = \dot{m}_1 h_{1,\text{in}} + \dot{m}_2 h_{2,\text{in}} \tag{3.19}$$

can be simplified by the relations of the constant mass flow rate as:

$$h_1 = h_{1,\text{in}} + h_{2,\text{in}} - h_2 . \tag{3.20}$$

By replacing the h_1 in Eq. (3.18) with Eq. (3.20), the first order differential equation can be derived for the enthalpy in channel 2 as:.

$$\frac{\partial h_2}{\partial z} + \frac{2W'_{1\to 2}}{\dot{m}_2} h_2 - \frac{W'_{1\to 2}}{\dot{m}_2} \left(h_{1,\text{in}} + h_{2,\text{in}} \right) = 0 . \tag{3.21}$$

By applying inlet condition at z=0, the analytic solution of the ch. 2 enthalpy with turbulent mixing model can be obtained as a function with exponential term as follows:

$$h_2 = \frac{1}{2} \left(h_{1,\text{in}} + h_{2,\text{in}} \right) + \frac{1}{2} \left(h_{2,\text{in}} - h_{1,\text{in}} \right) \exp\left(\frac{-2W'_{1\to 2}}{\dot{m}_2} z \right). \tag{3.22}$$

The analytic solution of ch. 1 can be derived in the same way as below:

$$h_1 = \frac{1}{2}\left(h_{1,\text{in}} + h_{2,\text{in}}\right) - \frac{1}{2}\left(h_{2,\text{in}} - h_{1,\text{in}}\right)\exp\left(\frac{-2W'_{1\to 2}}{\dot{m}_1}z\right). \qquad (3.23)$$

(2) Problem description

A test problem is designed to consist of two channels which are geometrically identical but have different inlet temperatures. The details of the geometrical information and the inlet boundary conditions are illustrated in Figure 3-6. The inlet temperature of channel 1 is set to be higher by 10°C than that of channel 2. The gap width is 0.003m and the effective mixing length is 0.0126m which is similar with a lattice in standard PWRs. The mixing parameter β is set to 0.035. The total axial height is 10.16m and the axial mesh size is 0.0254m (1 inch)

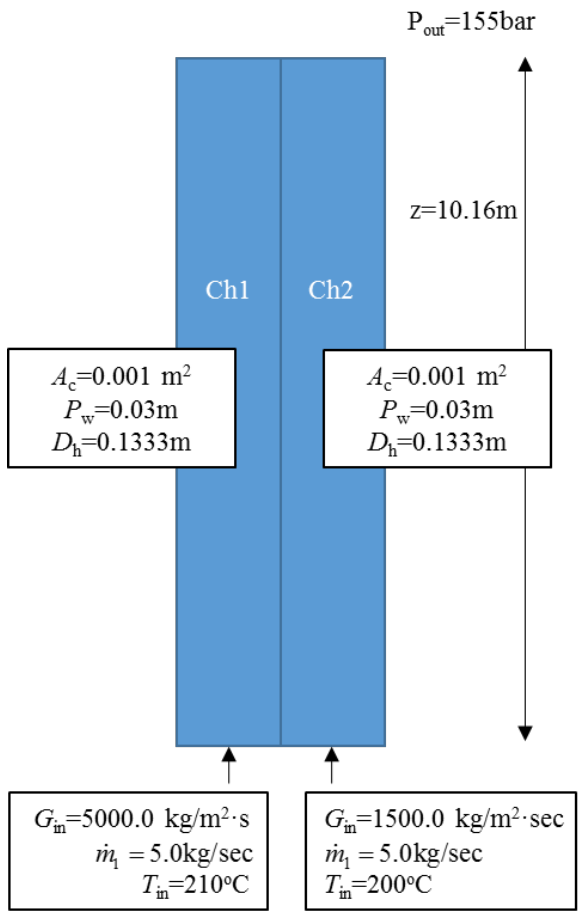

Figure 3-6. Geometry and inlet boundary condition of energy turbulent mixing test

(3) Result

The comparison between the analytic solution and the calculated results is drawn at Figure 3-7. The calculated result is matched well with the analytic solution, having the maximum relative error 1.19×10^{-4}% and the RMS of errors 6.21×10^{-5}%. As a result, it can be conclude that the turbulent mixing term in the energy equation is implemented properly.

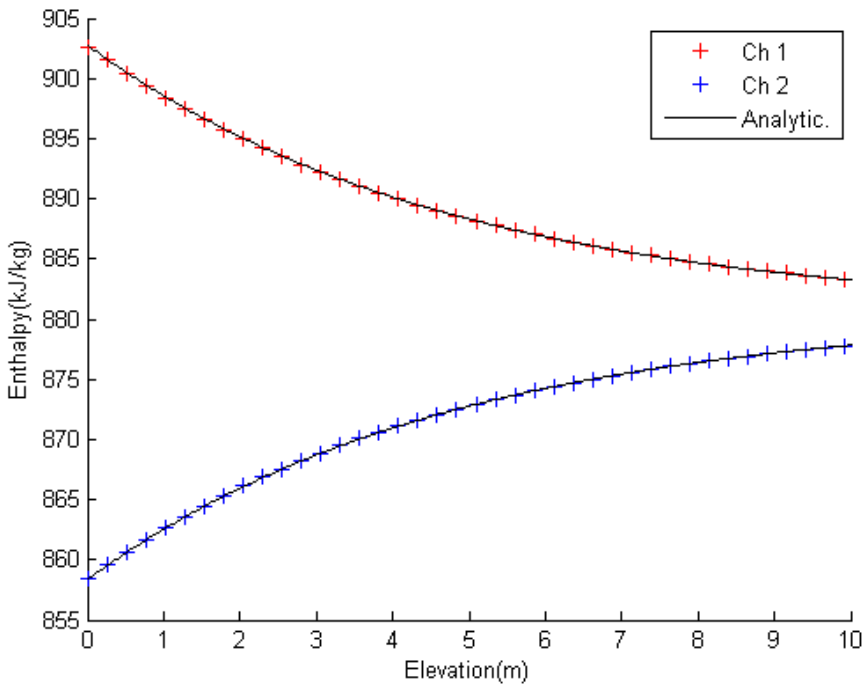

Figure 3-7. The calculated enthalpy by ESCOT and analytic solutions in turbulent mixing verification test

3.3. Validation

3.3.1. CNEN 4x4

The experiment has been performed by V. Marinlli *et al*. at Studsvik in 1972[57] to present 1) the results of flow redistribution between subchannels and 2) velocity profiles obtained under cold single-phase flow condition, and 3) a comparison with a subchannel code. Some of the experiment data are used for validation of ESCOT to verify following two mechanisms; those are the effect of turbulent mixing term in the momentum equation, and pressure drop by spacer grid.

The test section has a square-geometry with 16 (4x4) rods. It has a radial geometry as shown in Figure 3-8. The total height is 1.4m and a spacer grid is located at the middle of the test facility (0.7m). The experiment was performed with the 5 different inlet conditions: 0.64, 1.32, 2.61, 3.83, and 5.18 m/sec, and without any heat source. The outlet velocities at each subchannel were measured in the experiment.

The applied parameters in the constitutive relations during the calculation are written in Table 3-1. They include the wall friction factor, the form loss factor, and turbulent mixing parameter. The axial mesh is set to 0.028m by dividing the axial domain into 50 planes. Thus, 1,250 (5x5x50) meshes are used for the calculation. The outlet pressure and the inlet temperature are set to 0.10136 MPa and 300K respectively.

Table 3-1. Applied parameters in constitutive relations in CNEN 4x4

Type of model	**Model applied target**		**Value**
	Lateral momentum	Axial momentum	
Wall friction factor	-	O	$\mathrm{Re} < 2,300, \quad f = \frac{64}{\mathrm{Re}}$ $2,300 \le \mathrm{Re} < 3\times10^{4}, \quad f = 0.316\times\mathrm{Re}^{-0.25}$ $3\times10^{4} \le \mathrm{Re} < 1\times10^{6}, \quad f = 0.184\times\mathrm{Re}^{-0.20}$
Form loss factor	O	O	$K_{c,x} = 0.5$, $K_{c,z} = 0.3$ (Spacer grid)
Turbulent mixing parameter	-	O	$\beta = 0.2$

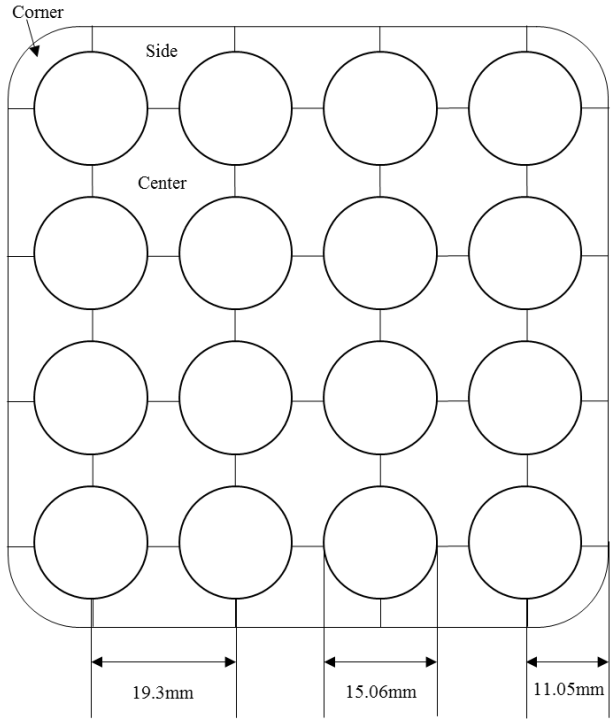

Figure 3-8. Cross section of test facility of CNEN 4x4

Figure 3-9 shows the pressure drops of case 1, 3, and 5 calculated by three different codes, MATRA[58], CUPID[59] and ESCOT. The experimental data measuring the pressure are not provided, so code-to-code validation is performed. All three codes predict the pressure drop by spacer grids, and the largest pressure drop is calculated at Case 5 which has the fastest inlet velocity. From the error comparison described in Table 3-2, it can be said that the results between three codes are well agreed by having 1.7% maximum difference. The maximum differences in each case are observed at the node where the spacer grid is located.

Table 3-2. Error of the calculated pressure of CNEN 4x4

	Case 1		**Case 3**		**Case 5**	
	Max Err (%)	RMS Err (%)	Max Err (%)	RMS Err (%)	Max Err (%)	RMS Err (%)
MATRA-ESCOT	0.048	0.028	0.462	0.065	1.669	0.234
CUPID-ESCOT	0.017	0.006	0.268	0.062	0.980	0.227

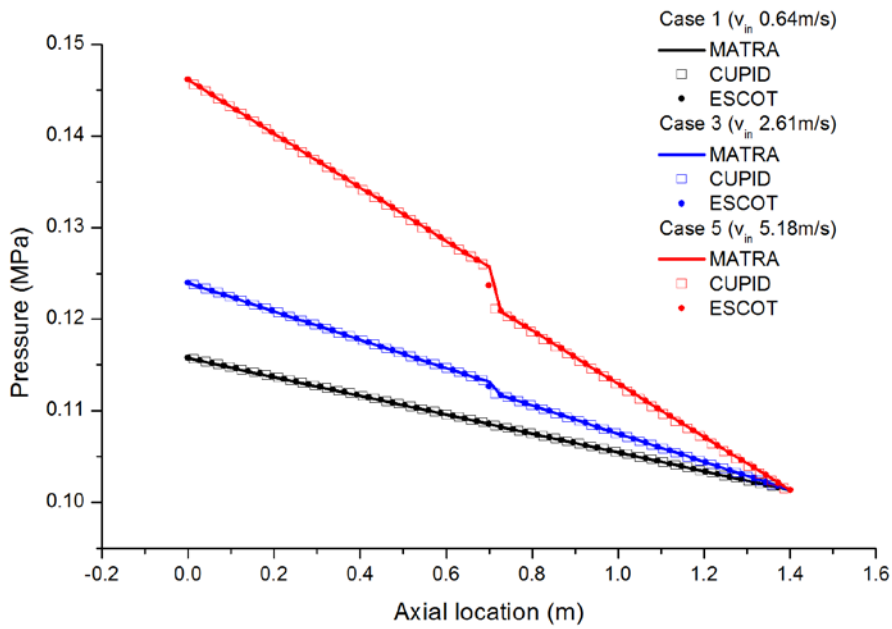

Figure 3-9. Pressure vs. axial elevation in CNEN 4x4 (Case 1, 3, 5)

The calculated outlet velocities are compared between three different codes as well as the measured data. By comparing results drawn in Figure 3-10 and Figure 3-11, one can figure out the effect of the turbulent mixing model. The corner cells which have high-resistance due to small hydraulic diameters lose their momentum more than other cells. The flow, instead, goes toward the center cells that have the lower resistance. In reality, this effect can be mitigated by the turbulent mixing mechanism. If the turbulent mixing model is disabled in the code, the lost momentum at the corner cells cannot be compensated, so the velocities at the corner are underestimated. In an opposite way, the velocities at the center cells are overestimated. The same tendency can be observed from the results of CUPID.

By adopting the turbulent mixing model, the code can capture a real phenomenon. The maximum error between the code and the measured date at the corner cells are reduced from 18% to 3.2%. In addition, it can be noticed from Table 3-3 that the predicted results from three codes show a good agreement by comparing errors with the measured data as the reference. In conclusion, the turbulent mixing model and the axial form loss term implemented in ESCOT are valid by this validation

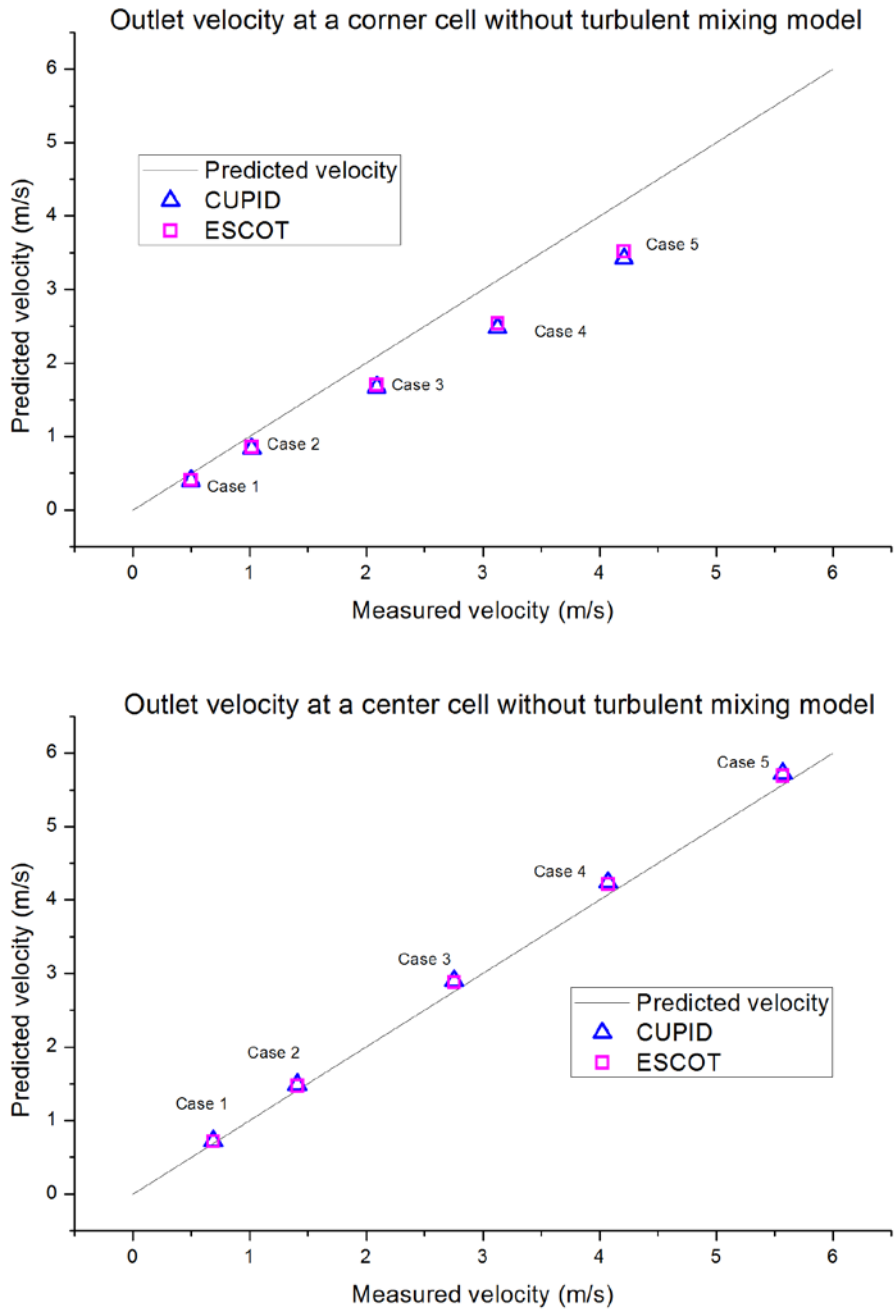

Figure 3-10. Calculated outlet velocities at a corner cell and center cell without turbulent mixing model

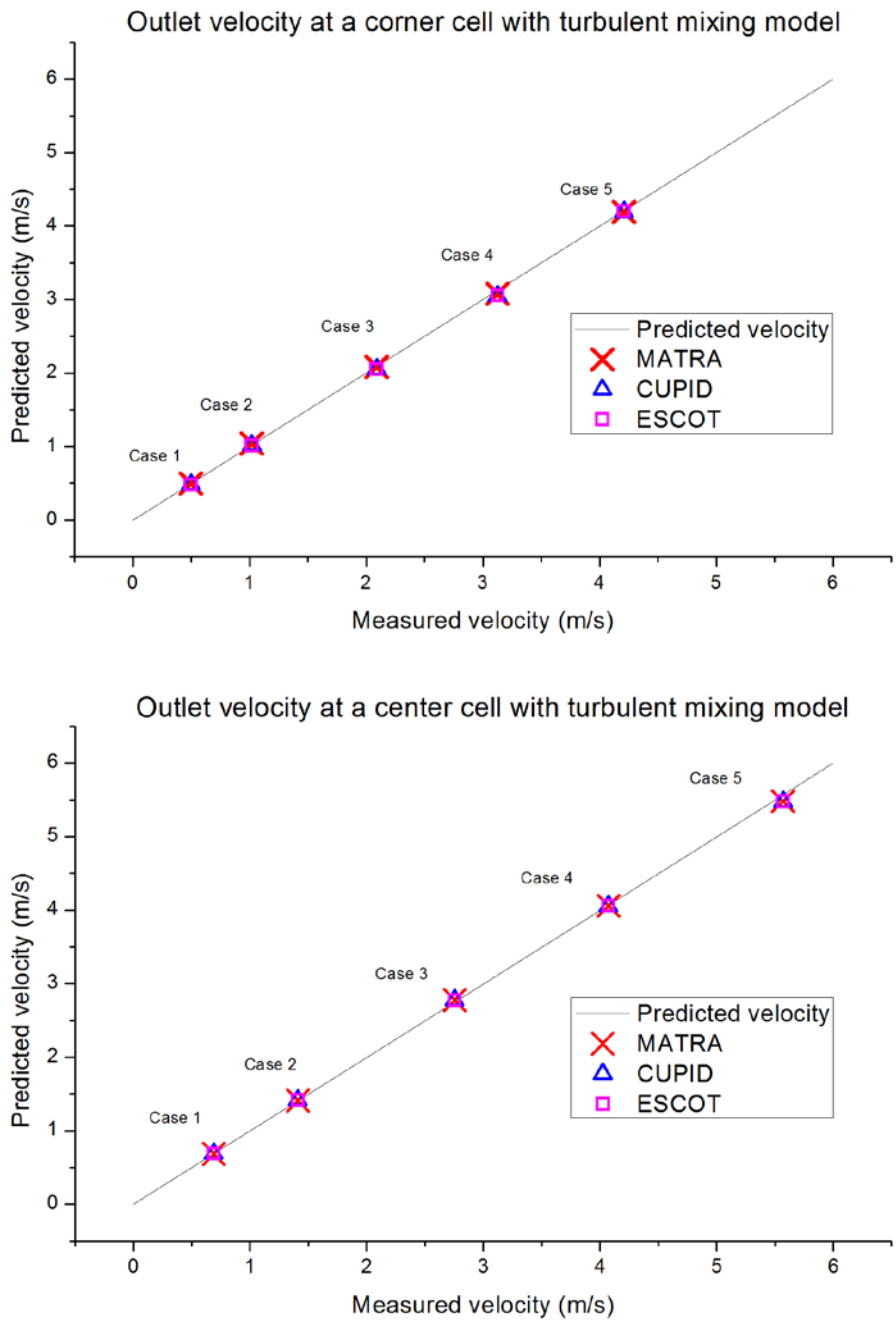

Figure 3-11. Calculated outlet velocities at a corner cell and center cell with turbulent mixing model

Table 3-3. Error of the calculated outlet velocities of CNEN 4x4

		Corner subchannel		Center subchannel	
		w/o turb. mixing	w/ turb. mixing	w/o turb. mixing	w/ turb. mixing
ESCOT	Max. Err (%)	18.71	3.17	4.69	1.66
	RMS Err (%)	17.50	1.94	3.89	0.85
CUPID	Max. Err (%).	20.44	3.02	5.54	1.67
	RMS Err (%)	19.27	1.91	4.74	0.92
MATRA	Max. Err (%).	-	2.21	-	1.60
	RMS Err (%)	-	1.22	-	0.85

3.3.2. WH 14x14

This experiment had been carried out by Chelemer et al. in 1973 at Westinghouse[60]. The purpose of the experiment was to observe the flow redistribution between two assemblies under an unheated condition when the flow blockage has arisen. Two types of the blockage were tested. One is a partial blockage that the inlet mass flow rate at the bundle 2 is half of that of the bundle 1. The other is a full blockage that the inlet mass flow rate at the bundle 2 is zero. As MATRA[58] and CUPID[59] did, the experiment is selected as validation test of ESCOT to check following aspects:

1) The capability to analyze the flow redistribution driven by the cross-flow effect
2) The capability to solve a problem which has reverse flow by the recirculation

The test section consists of two assemblies which have 14 x 14 unheated pins per assembly. The width and height of the cross section of the test section is 15.33 inch x 7.63 inch, and the axial elevation is 38 inch. The tube diameter is 0.426 inch and the Pitch to Diameter ratio (P/D) is 1.28. The flow rate is controlled by the valve installed at the in front of assembly inlet. A number of pitot tubes are installed to measure velocities and static pressure. Those are axially located in different 7 levels and at a certain level, 78 pitot tubes (26 x 3) are inserted. The detail geometry of the test section and locations of the pitot tubes are given at Figure 3-12 and Figure 3-13.

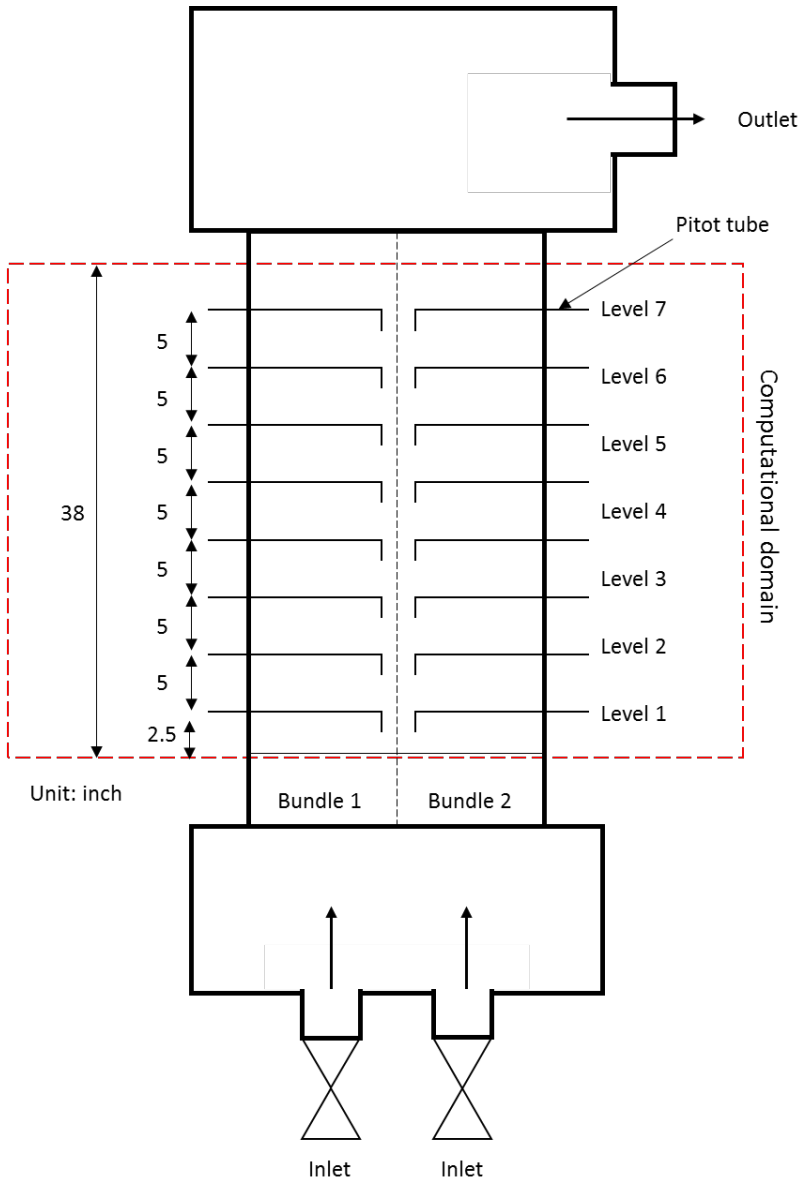

Figure 3-12. Longitudinal section of the WH 14x14 test section and locations of pitot tubes

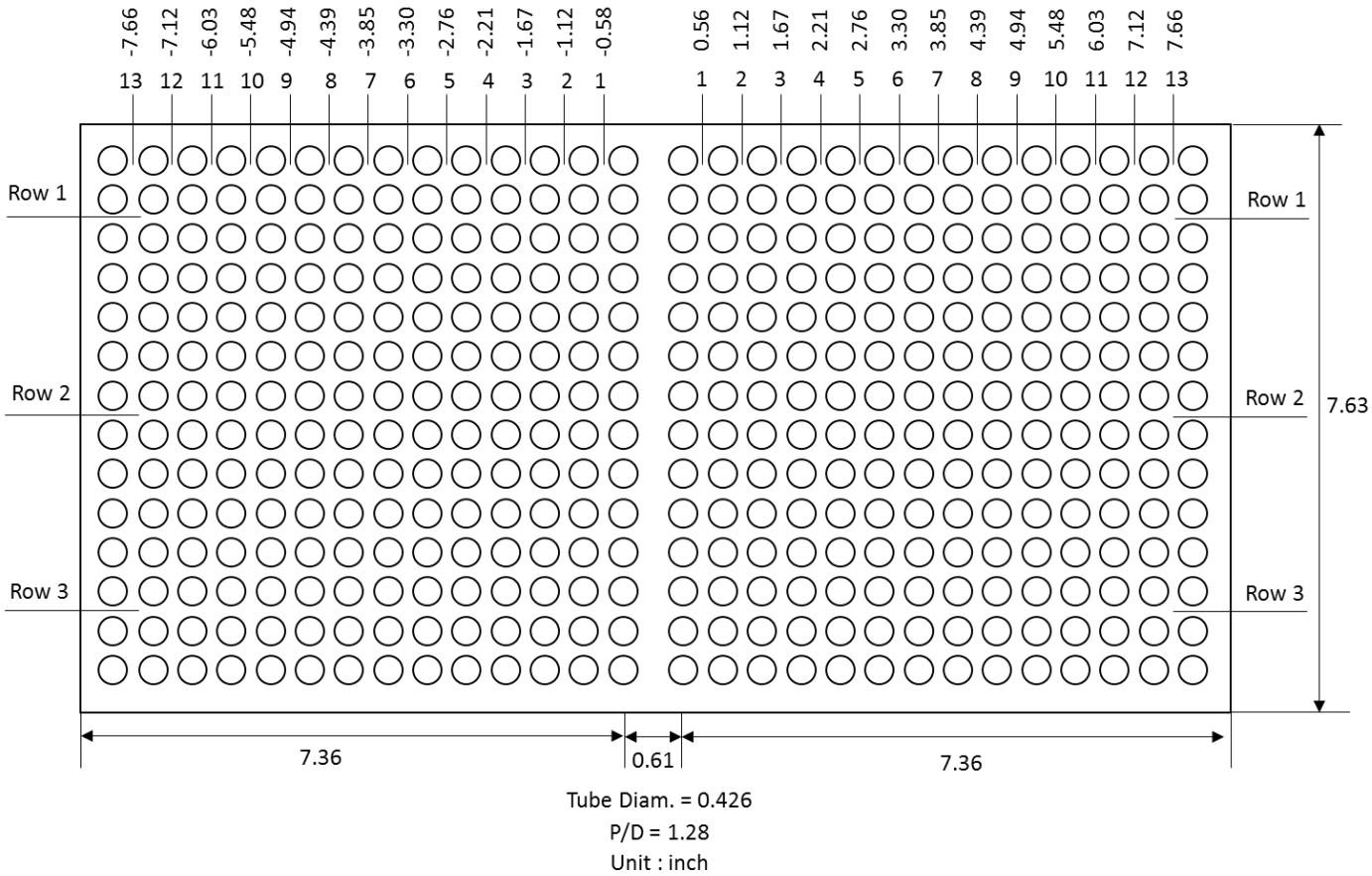

Figure 3-13. Cross-section of the WH test section and locations of pitot tubes

(1) Partial blockage case

The outlet pressure and the inlet temperature are set to 0.101MPa and 299.8K which the atmospheric condition assumed. The inlet mass flow rates are set to 1100gpm/550gpm in the partial blockage case. When the inlet mass flow rate is converted into velocities in SI unit, then they become 3.52m/sec and 1.76m/sec as the density is considered as 996.6 kg/m^3 at the given boundary condition. For the assembly gap channels, 2.64m/sec inlet velocity is assigned to consider that the inlet conditions of the channels are affected half and half.

The size of axial meshes is 1 inch (0.0254m), so the domain is divided into 38 planes. Thus, 16,530 meshes (29 x 15 x 38) are used in the calculation. The applied parameters in constitutive relations are given at Table 3-4. Note that turbulent mixing model is disables, so the crossflow is the only driving force for the flow redistribution.

Table 3-4. Applied parameters in constitutive relations in WH 14x14

Type of model	**Model applied target**		**Value**
	Lateral momentum	Axial momentum	
Wall friction factor	-	O	$f = \max[0.204 \times \mathrm{Re}^{-0.20}, 64.0 \times \mathrm{Re}^{-1.0}]$
Form loss factor	O	-	$K_{c,x} = 0.5$,
Turbulent mixing parameter	-	-	$\beta = 0.0$

As Figure 3-14 shows, the flow redistribution by the crossflow can be simulated by ESCOT. As the fluid moves up, the high axial velocities at the bundle 1 region become lowered, while the velocities become to be increased at the bundle 2 region. Figure 3-15 is another view which can observe the flow redistribution with respect to the portion of assembly averaged mass flow rate. The calculated results have the maximum relative error by 4.9% and the RMS of errors by 2.3% with experiment data as shown in Table 3-5. The maximum error appears at the top level. The magnitude of the error is similar as those of MATRA and CUPID.

Table 3-5. Error of the calculated portion of assembly averaged flow rate in WH 14x14 partial block case

	Portion of assembly averaged flow rate	
	Max Err (%)	RMS Err (%)
MATRA-EXP	5.399	2.412
CUPID-EXP	4.601	2.167
ESCOT-EXP	4.933	2.271

The local velocities at each measured level are compared between the results of codes and the experimental data. The predicted results and measured data are drawn at Figure 3-16. The velocities of row 1 to 3 are averaged, so the representative values at each level are calculated. Although the results of ESCOT are having the maximum relative error by 25.4% with experiment data, the tendency is agreed that the velocity profile changes from a skewed shape to an even shape as the axial elevation goes up.

When the only code-to-code comparison is considered as the results of ESCOT are set to the reference, the differences between codes are not significant as shown in Table 3-6. In particular, the results of ESCOT show the better agreement with those of CUPID. They all have a peak at the center of domain, and that is because the assembly gap channels have larger flow area than other channels.

Table 3-6. Error of the calculated local velocities in WH 14x14 partial block case

		Local velocity	
		Max Err (%)	RMS Err (%)
Code-to-Exp.	MATRA-EXP	29.9	12.8
	CUPID-EXP	27.4	12.3
	ESCOT-EXP	25.4	11.5
Code-to-Code	MATRA-ESCOT	7.35	3.82
	CUPID-ESCOT	2.24	1.29

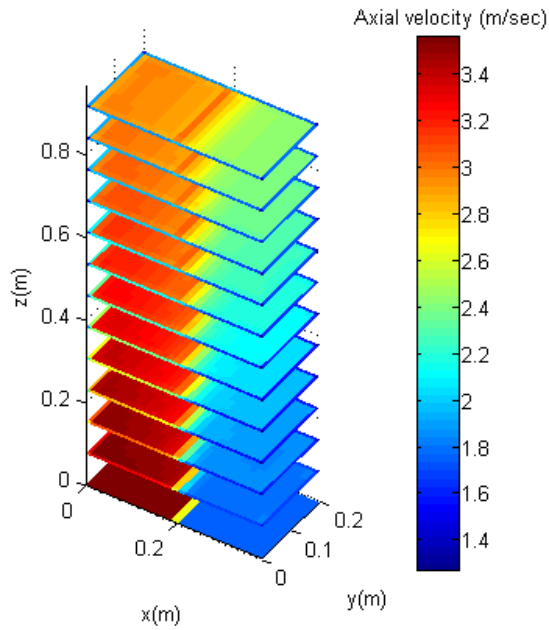

Figure 3-14. Axial velocity contour map of WH 14x14 partial blockage case

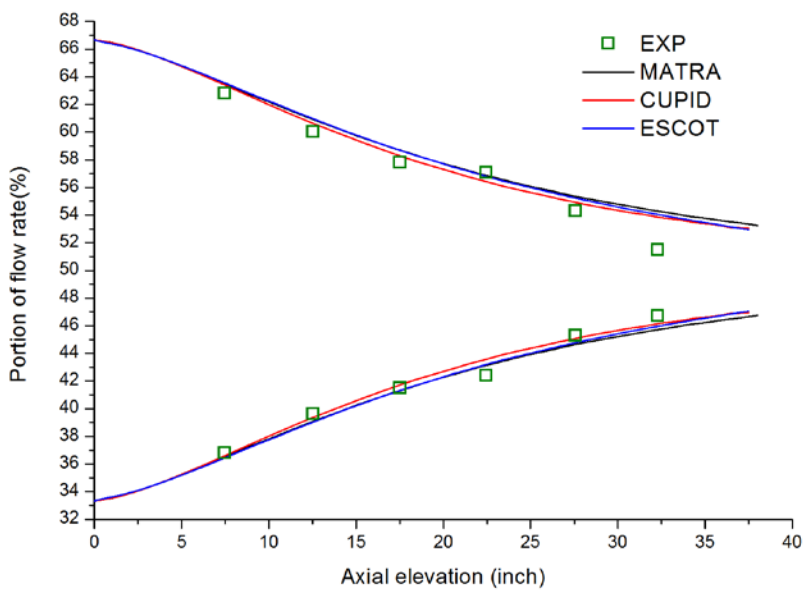

Figure 3-15. Portion of assembly averaged flow rate of WH 14x14 partial blockage case

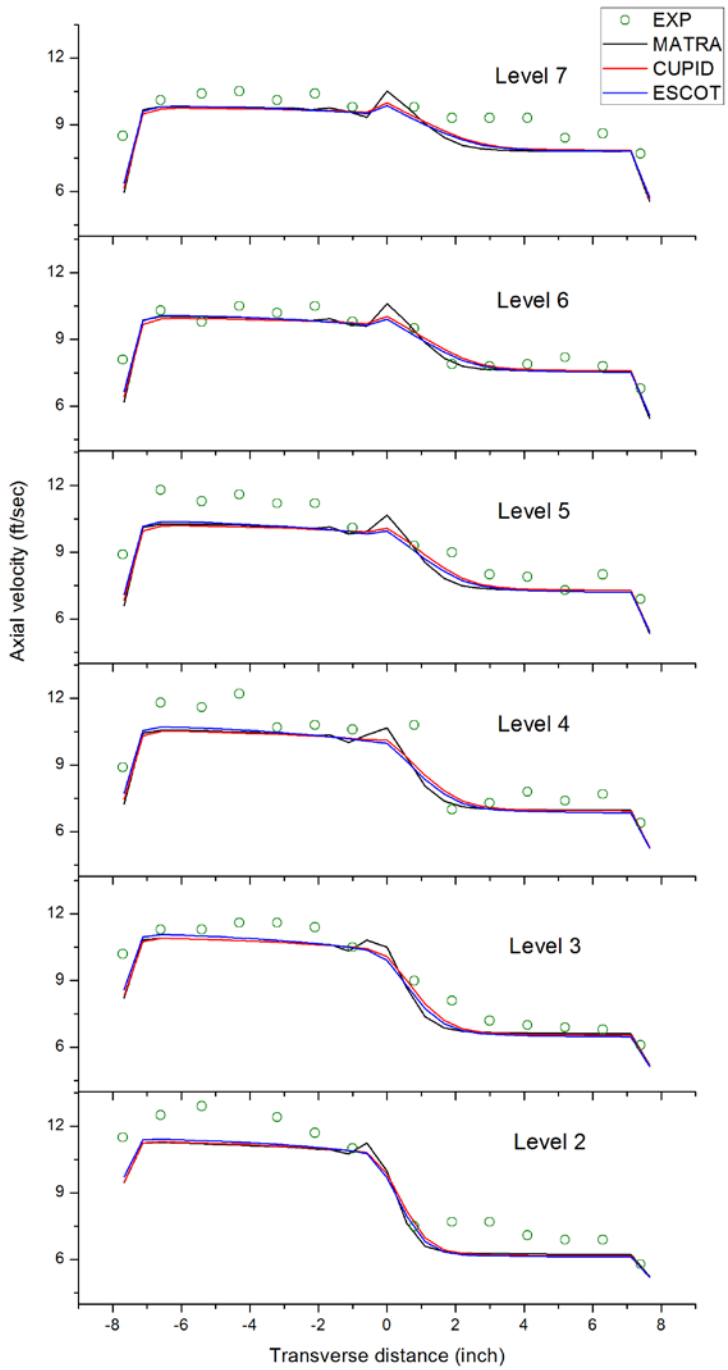

Figure 3-16. Row averaged axial velocity at measured levels of WH 14x14 partial blockage case

(2) Full blockage case

For the full blockage calculation, all of boundary conditions, meshing, and constitutive relations are remaining same except the inlet boundary condition. In this case, the inlet mass flow rates are set to 1100gpm/0gpm. When the inlet mass flow rate is converted into velocities in SI unit, then the inlet velocity at the bundle 1 and 2 becomes 3.52m/sec and 0.0m/sec respectively, and the one at the assembly gap channels is 1.76m/sec.

When the above inlet condition was initially given to the code without any treatment, the solutions were not converged due to large difference within time step. Therefore, a ramping boundary condition was applied to solve this case. The ramping boundary condition means that the uniform inlet boundary condition is initially given and gradually increases/decreases boundary conditions after the code reaches a certain stable state.

The flow redistribution is observed from Figure 3-17 as like the partial blockage case, but the negative axial velocities are detected at the low level planes, which does not exist in the partial blockage case. The recirculation can be seen more clearly with Figure 3-18, which is the vector map of velocity at the 2nd measuring row in the test section.

By solving the full blockage problem, it can be noted that a capability of analyzing the recirculation which has negative axial velocities depends on the numerical schemes. In case of the marching scheme, it solves the conservation equations in a certain plane, and the obtained solutions will be used as boundary conditions at the next plane. It has advantages that the solution can easily satisfy conservation equation and save computing time because the small subdomains are handled during the calculation. However, if there is a reverse flow, then the boundary condition for the next plane cannot be defined, so special treatments are required such as applying an explicit scheme. On the other hands, the pressure-velocity linked schemes may take longer computing time than the marching scheme due to large dimension of the matrix. However, it solves the problem to satisfy conservation equation over the whole domain, so the local reverse flow can be easily solved.

Only the results of CUPID were available for the full blockage case, so the results are compared only between ESCOT and CUPID. The RMS of the relative errors of local velocities at level 1, 2, and 7 is 72.9% as the result of CUPID is the reference, but the large difference mainly results from the different sign and the small magnitude of velocities in the recirculation region. As Figure 3-19 shows, ESCOT

and CUPID have a same tendency by having the reverse flow at level 1 and 2 and the redistributed flow profile at level 7. In the end, ESCOT is verified to have the capability to simulate the flow redistribution by crossflow and have a capability to solve reverse flow as well.

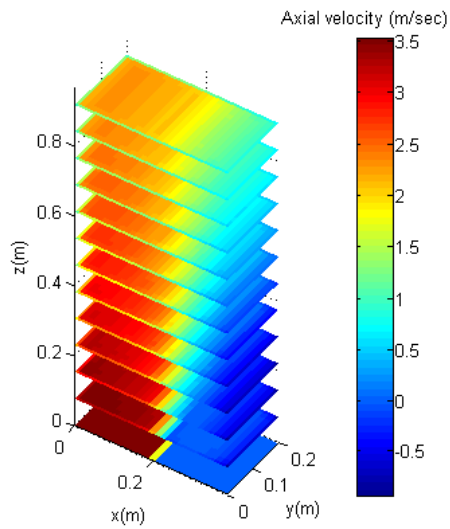

Figure 3-17. Axial velocity contour map of WH 14x14 full blockage case

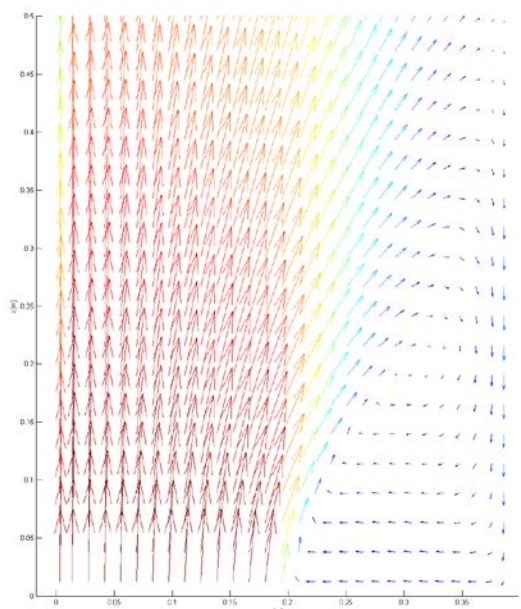

Figure 3-18. Velocity vector map at the 2nd measuring row line of WH 14x14 full blockage case

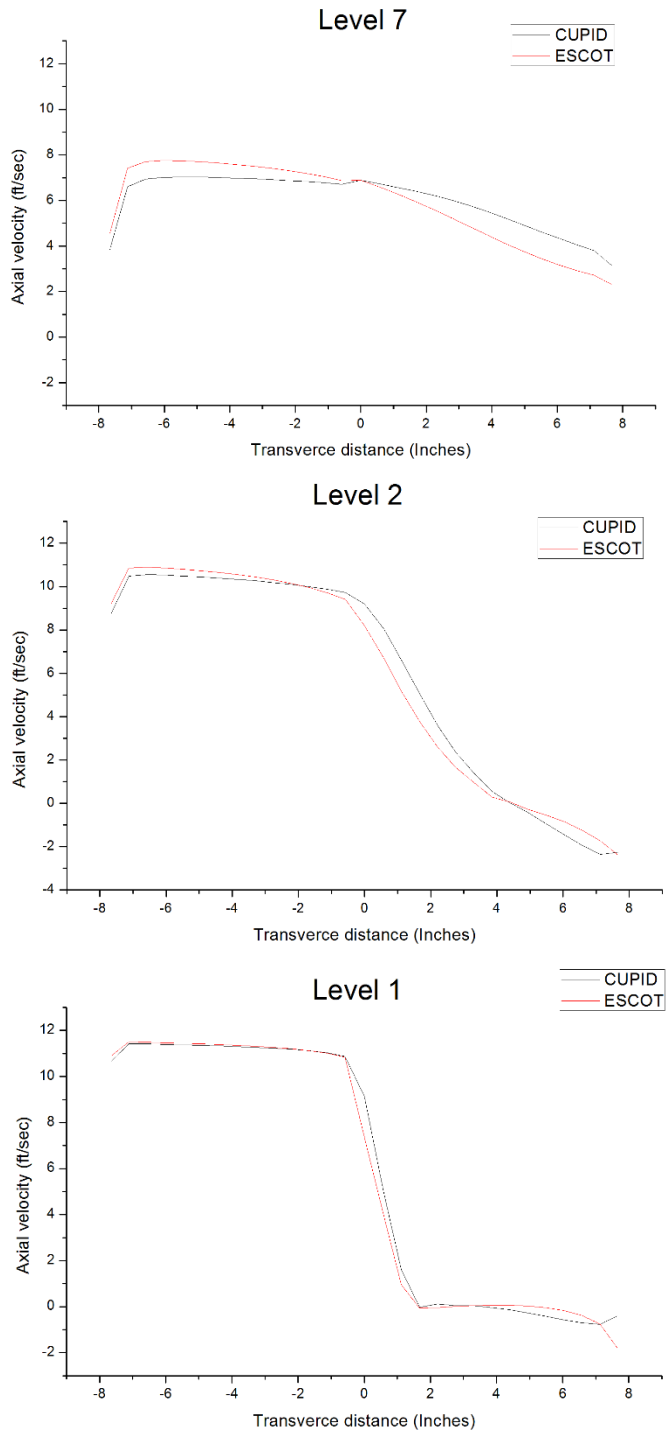

Figure 3-19. Averaged axial velocity at measured levels 1, 2, and 7 of WH 14x14 full blockage case

3.3.3. 9x9 mini MSLB problem

Because the previous validation problems are all unheated cases, a heated validation problem is designed. The calculated results from three different codes, CUPID, MATRA and ESCOT are compared as the code-to-code validation to check following features:

1) Capability to obtain right solutions under an asymmetric inlet condition and checkerboard-shape-power distribution
2) Computing time comparison: velocity-pressure coupled algorithm VS. marching scheme; two-fluid model based VS. drift-flux model based.

The problem is designed to have a 9x9 array of fuel assemblies which is a quarter-core-size and severe asymmetric temperature profile which can be similar when the Main Steam Line Break (MSLB) accident occurs. When the main steam line is broken, the supercooling at the primary side might occur in the steam generator by the sudden evaporation at the secondary side. The cold water in the U-tube is injected to the core by the pump-driven-circulation, and this leads the negative feedback corresponding to the Moderator Temperature Coefficient (MTC). In addition, if the control rod is failed occasionally at the position where the cold water is injected, then the recriticality will happen by the positive reactivity. The inlet conditions and power distribution of the problem are set to simulate the MSLB accident. The input parameters are given at Table 3-7.

The calculated temperature distributions from ESCOT and CTF are drawn at Figure 3-20. The overall shape of the profile is similar between two codes. The bundle averaged values including enthalpy, pressure and mass flow rate are illustrated in Figure 3-21. The discrepancy of the bundle average is given as RMS difference and maximum relative difference in Table 3-8. The results of ESCOT are set as the reference. According to Table 3-8, the results of ESCOT have the better agreement with those of MATRA, but all results are matched well within 0.06% difference. The bundle pressure of CTF is slightly higher than that of other codes, because defining outlet boundary pressure is different. In case of ESCOT and MATRA, the outlet boundary pressure is defined at the surface of the last axial nodes, while CTF defines it at the ghost cells locating above the last axial plane.

Table 3-7. Input parameters of 9x9 mini MSLB problem

Parameters		**Values**
Boundary & initial condition	Inlet temp.	Half and half of a core: 295.83 ºC / 265.83 ºC
	Outlet pressure	15.513 MPa
	Total inlet mass flow rate	7023.753 kg/sec
Geometry information	# of Pins in a row per assembly	16
	Pin diameter	9.5 mm
	Pin pitch	12.8776 mm
	Assembly pitch	208.756 mm
	Active length	3.810 m
	# of guide tubes per assembly	20 (4×5)
	# of total meshes	841,000 (145eshes40)
Power information	Assembly power distribution	Nominal power: 15.904 MW Checkerboard loading: 1.2/0.8 One assembly has 2.0 nominal power where the cold water is injected.
	Pin power distribution	Axial: Uniform Radial: Uniform except guide tube having zero
Parameters for Constitutive relations	Turbulent mixing	$\beta = 0.05$
	Form loss	Kx=0.5 (only for gap)
	Pressure drop	$f = \max[64/\mathrm{Re},\ 0.204\,\mathrm{Re}^{-0.2}]$

For the mass flow rate, the result of MATRA is the best among three, which should be constant along the axial direction. The best result of MATRA comes from the marching scheme. The velocity-pressure coupled algorithm solves the pressure correction equation over the whole domain to satisfy the continuity, whereas the marching scheme solves conservation equations in one plane and moves toward next plane. The continuity can be satisfied more easily in the smaller geometry, so the marching scheme can result in the better solutions for mass flow rate.

The computing time of the single processor calculation① is compared between three codes, as shown at Table 3-9. The t in the table indicates the problem time. The calculation in CTF is finished at t=1.4 sec by pseudo steady-state conditions. The ESCOT result at 2nd column is when the calculation is terminated by satisfying steady-state conditions, and the one at 3rd column is when the calculation is finished by reaching the maximum problem time as same in that of CTF. The same convergence criteria for steady-state were set to CTF and ESCOT, which are the mass and energy storage by 0.5% and the mass and energy balance by 0.01%. In case of MATRA, the SCHEME solver was used, which is steady-state solver, so the computing time to obtain the converged solutions is only given.

From the result, it can be concluded that the choice of the drift-flux model can save the computing time when it is compared with the two-fluid three field model based code. In particular, time reduction at solving momentum equations is significant, and it is major advantage of the drift-flux model by taking one mixture momentum equation instead of phasic momentum equations. The marching scheme comes up with the shortest computing time by the results of MATRA. It suggests that the combination of the marching scheme and the HEM can effectively reduce computational burden in subchannel analyses.

① Intel(R) Xeon (R) CPU E5-2640 v3 @ 2.60GHz

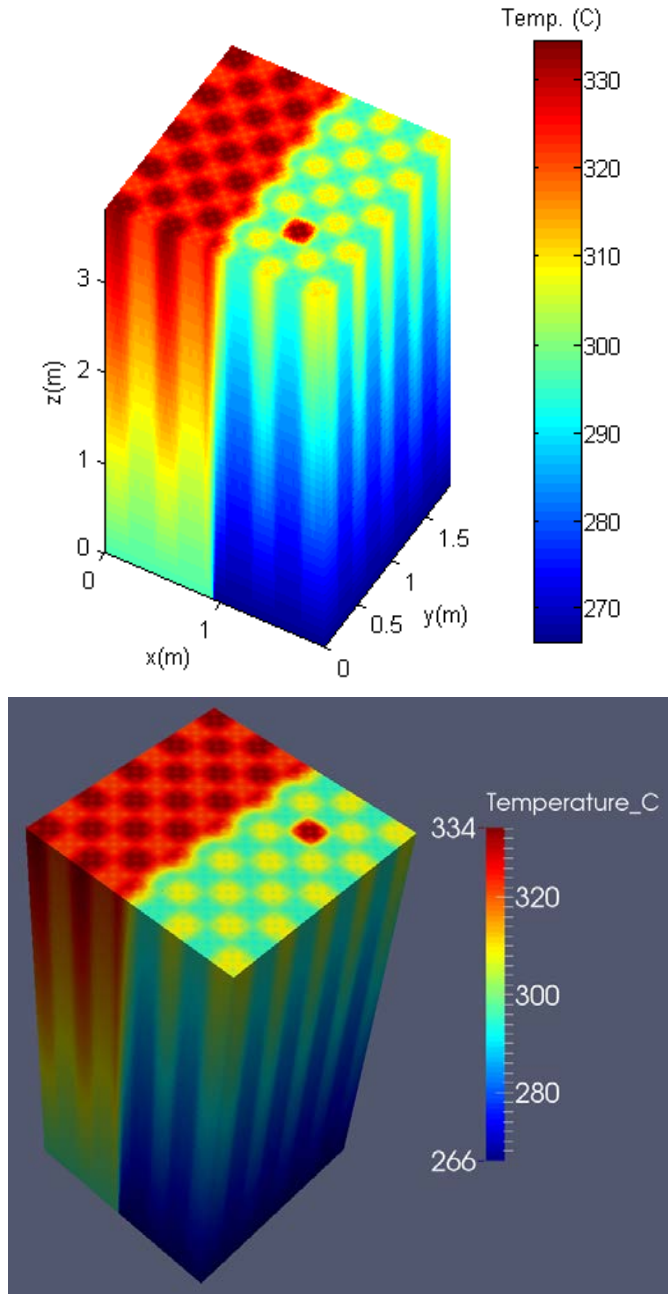

Figure 3-20. Temperature profile of 9x9 mini MSLB problem (up: ESCOT, down: CTF)

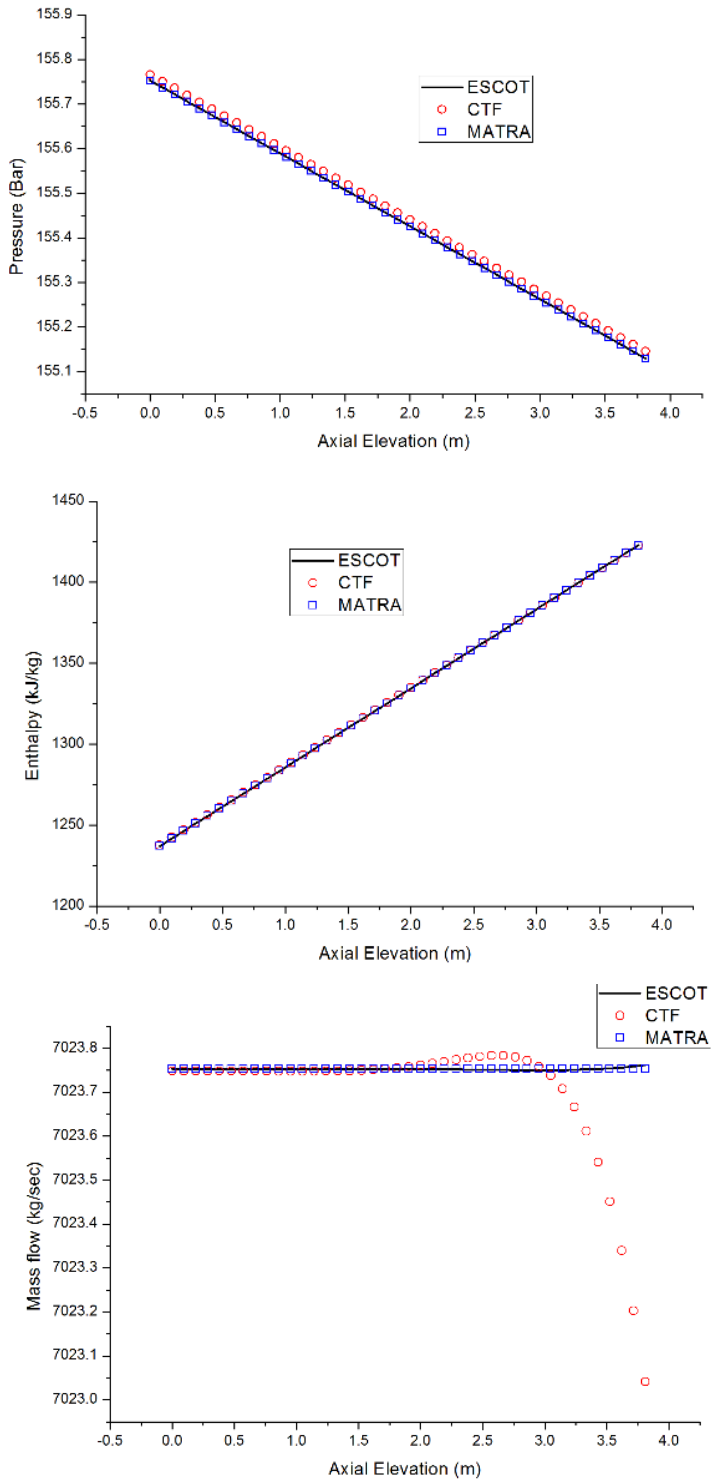

Figure 3-21. Bundle average results of 9x9 MSLB calculation

Table 3-8. Discrepancy of bundle average of 9x9 MSLB calculation

	Pressure		**Enthalpy**		**Mass flow rate**	
	CTF-ESCOT	MATRA-ESCOT	CTF-ESCOT	MATRA-ESCOT	CTF-ESCOT	MATRA-ESCOT
RMS diff (%)	0.0094	3.07E-4	0.0348	0.0117	0.0024	2.95E-5
Max. rel. diff (%)	0.0101	3.99E-4	0.0612	0.0223	0.0103	1.32E-4

Table 3-9. Computing time of 9x9 MSLB calculation

	CTF (t=1.4sec)	**ESCOT (t=1.06sec)**	**ESCOT (t=1.4 sec)**	**MATRA**
Total CPU time	2h 42min	41 min	53 min	7 min 42sec
Solving pressure Eq.	1h 07min	27 min	36 min	-
Setup and solve momentum Eq.	45 min	51 sec (setup is not considered)	1 min 05 sec (setup is not considered)	3 min
Setup and solve continuity Eq.	13 min	3 min 03sec (setup is not considered)	4 min 05sec (setup is not considered)	12 sec
Matrix and source update	-	8min	11min	-
Outer iteration	281	103	138	80

Chapter 4. Investigation of Efficient Solvers for Linear System Involving Pressure Correction Matrix

4.1. Problems on Solving Pressure Correction Matrix

Compared to SMAC- or ICE- type algorithms, the SIMPLE-like algorithms may suffer from a computational burden, because more linear systems should be solved in a one outer iteration. Solving large linear system is an expensive process in aspect to computation, and the efficiency of codes may be determined by a computing time spent for solving them in many cases. Therefore, choosing the efficient linear solver is important.

The direct solution methods such as Gauss elimination and LU factorization are basic ones to solve linear systems. They are the most accurate ways to obtain solutions of linear systems. However, in case of sparse matrix linear systems, it induces fill-in that results in significant computational burden. For instance, a test problem is designed to check how much computing portion is taken by solving linear systems with a direct solver. It has 16 x 16 pins, single assembly geometry including guide tubes. The radially and axially uniform power is assigned to the problem. Detail parameters are shown at Table 4-1. For the direct solver, SuperLU[21] is used, that is an open library pack to provide the direct solution of large, sparse, nonsymmetric systems of linear equations on high performance machines. As shown in Table 4-2, 97% of total computing time was spent for solving momentum and pressure linear systems. Thus, it is necessary to adopt more efficient linear solvers.

The iterative solution methods are well known as an efficient ways to deal with sparse linear systems. The solutions of iterative solution methods are not exact ones of the original linear system, but the error of solutions is diminished as the iteration goes on. The iteration will be stopped when the error is less than a certain criterion. High reduction rate or convergence rate is crucial with respect to the efficiency of iterative solution methods.

Table 4-1. Input parameters for the single assembly problem

Parameters		Values
Boundary & initial condition	Inlet temp.	295.83 °C
	Outlet pressure	15.513 MPa
	Total inlet mass flow rate	85.873 kg/sec
Geometry information	# of Pins in a row per assembly	16
	Pin diameter	9.5 mm
	Pin pitch	12.8776
	Assembly pitch	208.756 mm
	Active length	3.810m
	# of guide tubes per assembly	20 (4×5)
	# of total meshes	11,560 (17 meshes
Power information	Assembly power distribution	15.904 MW
	Pin power distribution	Axial: Uniform Radial: Uniform except guide tube having zero
Parameters for Constitutive relations	Turbulent mixing	$\beta = 0.05$
	Form loss	Kx=0.5 (only for gap)
	Pressure drop	$f = \max[64/\mathrm{Re},\ 0.204\,\mathrm{Re}^{-0.2}]$

Table 4-2. Computing time with SuperLU of 1x1 assembly test problem

Type of solution process	Computing time (sec)	Portion of total computing time (%)
Initialization	0.9	0.3
Matrix elements update	1.9	0.6
Source update	0.7	0.2
Solving momentum eqns.	159.3	49.1
Solving pressure eqns.	156.3	48.2
Solving scalar eqns.	1.9	0.6
Total computing time	324.7	100

The convergence rate of the traditional stationary iterative methods, such as Successive Over-Relations (SOR), is determined by spectral radius of iteration matrix as defined as:

$$\rho(T) = \max_i \left(|\lambda_i| \right) \tag{4.1}$$

where T: iteration matrix

λ: Eigenvalue of iteration matrix.

If the spectral radius is smaller, the iterative method will have a high convergence rate. In addition, the stationary iterative methods are converged only when the spectral radius is less than one. Or it can be determined by a diagonal dominance of an original matrix. A matrix can be diagonally dominant when its elements satisfy:

$$d_i \geq \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}| \tag{4.2}$$

where d_i : diagonal element at *i*-th row of matrix

a_{ij} : element of matrix at *i*-th row and *j*-th column.

If the diagonal element is much larger than the sum of absolute value of off-diagonal elements, then the matrix has a strong diagonal dominance and simultaneously, it has high convergence rate.

Now, let us consider the elements of pressure matrix in Eq.(2.88). The pressure matrix intrinsically has a diagonal dominance because a diagonal term is larger than the sum of off-diagonal elements by one. However, the magnitude of off-diagonal terms is usually 10 to the 4 or 6 which is much bigger than one, so the diagonal dominance of pressure matrix is weak. In this reason, many traditional iterative solvers show poor performance to solve the linear system involving the pressure matrix.

Many researches tried to find a proper linear solver for the pressure equation, and it turned out that the Krylov Subspace Methods (KSM) can be one of the best choices[23,24]. Thus, two KSMs, BiCGSTAB and GMRES, are adopted as linear solvers in ESCOT. In the following section, the theoretical background and algorithms of the KSMs are briefly introduced. The performance test of various linear solvers with a set of sample tests is presented as well.

4.2. Introduction of Krylov Subspace Method

The *k*-order Krylov subspace of an n-by-n matrix **A** is given by:

$$K_k = \left\{ \mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \mathbf{A}^2\mathbf{r}_0, \cdots, \mathbf{A}^{k-1}\mathbf{r}_0 \right\}, \quad k \le n. \tag{4.3}$$

The arbitrary vector $\mathbf{r}_0$ is normally taken to an initial residual vector of a linear system $\mathbf{Ax}=\mathbf{b}$, which is $\mathbf{r}_0=\mathbf{b}-\mathbf{Ax}_0$. The concept of Krylov subspace methods for solving linear systems is to find the solutions of the linear system using the spanned subspace and iterations.

4.2.1. BiCGSTAB

The Conjugate Gradient (CG) method[61] which was proposed in 1950s is the origin of the Krylov subspace methods. The method was derived to solve the symmetric positive definite linear systems. Let us define a functional as below:

$$F(\mathbf{x}) = \frac{1}{2}\langle \mathbf{x}, \mathbf{Ax} \rangle - \langle \mathbf{x}, \mathbf{b} \rangle. \tag{4.4}$$

The functional has the minimum where $x=A^{-1}b$. The CG algorithm is constructed to find a solution vector which satisfies the minimum point of the functional.

Because the applicant of the original CG method is restricted to the symmetric positive definite linear systems, many variants of the algorithm were proposed to deal with nonsymmetric linear systems. The BiConjugate Gradient Stabilized (BiCGSTAB)[27] method is one of them, and it was proved that the BiCGSTAB algorithm has the better convergence stability than other CG-like algorithms. The numerical algorithm of the BiCGSTAB method is given at Table 4-3.

Table 4-3. BiCGSTAB algorithm

$\mathbf{r}_0 = \mathbf{b} - \mathbf{Ax}_0$ for an initial guess $\mathbf{x}_0$;
$\rho_0 = \alpha = \omega_0 = 1$; $\mathbf{v}_0 = \mathbf{p}_0 = 0$
For *i*=1,2,3
$\rho_i = (\mathbf{r}_0, \mathbf{r}_{i-1})$; $\beta = (\rho_i / \rho_{i-1})(\alpha / \omega_{i-1})$; $\mathbf{p}_i = \mathbf{r}_{i-1} + \beta(\mathbf{p}_{i-1} - \omega_{i-1}\mathbf{v}_{i-1})$
$\mathbf{v}_i = \mathbf{Ap}_i$
$\alpha = \rho_i / (\mathbf{r}_0, \mathbf{v}_i)$; $\mathbf{s} = \mathbf{r}_{i-1} - \alpha\mathbf{v}_i$
$\mathbf{t} = \mathbf{As}$
$\omega_i = (\mathbf{t}, \mathbf{s}) / (\mathbf{t}, \mathbf{t})$; $\mathbf{x}_i = \mathbf{x}_{i-1} + \alpha\mathbf{p}_i + \omega_i\mathbf{s}$
If $\mathbf{x}_i$ is converged, then quit
$\mathbf{r}_i = \mathbf{s} - \omega_i\mathbf{t}$
End

4.2.2. GMRES

Instead of minimizing the functional in CG-like methods, the Generalized Minimal Residual (GMRES)[26] approximates the solution as the form $\mathbf{x}_0+\mathbf{z}_m$ which minimizes the residual norm, $\mathbf{r}=\mathbf{b}-\mathbf{A}\mathbf{x}$, over $\mathbf{z}_m$ in the *m*-order Krylov subspace. It can be expressed as below:

$$\min_{z_m \in K_m} \left\| \mathbf{b} - \mathbf{A}\mathbf{x}^* \right\| = \min_{z_m \in K_m} \left\| \mathbf{b} - \mathbf{A}\left(\mathbf{x}_0 + \mathbf{z}_m\right) \right\| = \min_{z_m \in K_m} \left\| \mathbf{r}_0 - \mathbf{A}\mathbf{z}_m \right\| \tag{4.5}$$

where $\mathbf{x}^*$ is a approximated solution

$\mathbf{A}$ is a n-by-n dimension matrix.

Because vector $\mathbf{z}_m$ is in the Krylov subspace, it can be represented with the form of superposition of orthogonal vectors as:

$$\mathbf{z}_m = \mathbf{v}_1 y_1 + \mathbf{v}_2 y_2 + ... + \mathbf{v}_m y_m = \mathbf{V}_m \mathbf{y}_m \tag{4.6}$$

where $\mathbf{V}_m = \left[\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_m\right] \in \mathbf{R}^{n,m}$

$\mathbf{v}_i$: A Column orthonormal vector $\in \mathbf{R}^{m,1}$.

The set of orthonormal bases of Krylov subspaces can be obtained with the Arnoldi's process[62]. The algorithm is described at Table 4-4. By using Eq.(4.6), Eq.(4.5) is changed as follows:

$$\min_{z \in K_k} \left\| \beta \mathbf{v}_1 - \mathbf{A}\mathbf{V}_m \mathbf{y}_m \right\| \quad \text{where} \quad \beta = \left\| r_0 \right\|. \tag{4.7}$$

Table 4-4. Arnoldi's process

Choose $\mathbf{x}_0$ and compute $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}$ and $\mathbf{v}_1 = \dfrac{\mathbf{r}_0}{\left\| \mathbf{r}_{.0} \right\|}$

% Arnoldi Process in Krylov Subspace

For *j*=1:m

$h_{ij} = \left\langle \mathbf{A}\mathbf{v}_j, \mathbf{v}_i \right\rangle$ for *i*=1,2….,*j*

$\tilde{\mathbf{v}}_{j+1} = \mathbf{A}\mathbf{v}_j - \sum_{i=1}^{j} h_{ij}\mathbf{v}_i$

$h_{j+1,j} = \left\| \tilde{\mathbf{v}}_{j+1} \right\|$

$\mathbf{v}_{j+1} = \tilde{\mathbf{v}}_{j+1} / h_{j+1,j}$

End

The coefficient h_{ij} can consist of the Hessenberg matrix as follows:

$$\mathbf{H}_m = \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1m-1} & h_{1m} \\ h_{12} & h_{22} & & h_{2m-1} & h_{2m} \\ 0 & h_{32} & \ddots & h_{3m-1} & h_{3m} \\ \vdots & 0 & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & h_{mm-1} & h_{mm} \end{bmatrix} \in \mathbf{R}^{m,m}$$

and

$$\bar{\mathbf{H}}_m = \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1m-1} & h_{1m} \\ h_{21} & h_{22} & & h_{2m-1} & h_{2m} \\ 0 & h_{32} & \ddots & h_{3m-1} & h_{3m} \\ \vdots & 0 & \ddots & \ddots & \vdots \\ & & 0 & h_{mm-1} & h_{mm} \\ 0 & & & 0 & h_{m+1m} \end{bmatrix} \in \mathbf{R}^{m+1,m} . \qquad (4.8)$$

From the algorithm of Arnoldi's process, a relation between the Hessenberg matrix, orthonormal vectors and original matrix **A** can be formulated as follows:

$$\begin{aligned} \mathbf{A}\mathbf{v}_j &= \sum_{i=1}^{j+1} h_{ij}\mathbf{v}_i \\ \mathbf{A}\mathbf{V}_m &= \mathbf{V}_{m+1}\bar{\mathbf{H}}_m \end{aligned} . \qquad (4.9)$$

Let us define $J(\mathbf{y}_m) \equiv \left\| \beta\mathbf{v}_1 - \mathbf{A}\mathbf{V}_m\mathbf{y}_m \right\|$ in Eq.(4.7). $J(\mathbf{y}_m)$ can be rewritten by using $\mathbf{v}_1 = \mathbf{V}_{m+1}\mathbf{e}_1^{(m+1)}$ and Eq.(4.9) as:

$$J(\mathbf{y}_m) = \left\| \left(\beta\mathbf{e}_1^{(m+1)} - \bar{\mathbf{H}}_m\mathbf{y}_m \right) \right\| \qquad (4.10)$$

where $\mathbf{e}_1^{(m+1)} = [1,\ 0,\ ..,\ 0]^T \in \mathbf{R}^{m+1}$.

In the end, the final goal of the GMRES algorithm becomes to find $\mathbf{y}_m$ which satisfies $\min \left\| \left(\beta\mathbf{e}_1^{(m+1)} - \bar{\mathbf{H}}_m\mathbf{y}_m \right) \right\|$ by using least square method. To do this, let us introduce plane rotation matrices to transform the Hessenberg matrix into upper triangular form. The plane rotation matrix which rotates a unit vector $\mathbf{e}_i$ on plane to x-axis by angle θ can be defined as follows:

$$\boldsymbol{\Omega}_i = \begin{bmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & 1 & & & & \\ & & & c_i & s_i & & \\ & & & -s_i & c_i & & \\ & & & & & 1 & \\ & & & & & & \ddots \\ & & & & & & & 1 \end{bmatrix} \in \mathbf{R}^{m+1,m+1} \text{ with } c_i^2 + s_i^2 = 1. \quad (4.11)$$

The first plane rotational operation changes the elements of the matrix and vector of $J(\mathbf{y}_m)$ as follows:

$$\boldsymbol{\Omega}_1 \bar{\mathbf{H}}_m = \begin{bmatrix} h_{11}^{(1)} & h_{12}^{(1)} & \cdots & h_{1m-1}^{(1)} & h_{1m}^{(1)} \\ & h_{22}^{(1)} & & h_{2m-1}^{(1)} & h_{2m}^{(1)} \\ & h_{32} & \ddots & h_{3m-1} & h_{3m} \\ & & \ddots & \ddots & \vdots \\ & & & h_{m-1m-1} & h_{mm} \\ & & & & h_{m+1m} \end{bmatrix} \text{ and } \boldsymbol{\Omega}_1 \beta \mathbf{e}_1^{(m+1)} = \begin{bmatrix} c_1\beta \\ -s_1\beta \\ 0 \\ \vdots \\ \\ 0 \end{bmatrix} \quad (4.12)$$

where $\boldsymbol{\Omega}_1 = \begin{bmatrix} c_1 & s_1 & & & & \\ -s_1 & c_1 & & & & \\ & & 1 & & & \\ & & & 1 & & \\ & & & & \ddots & \\ & & & & & 1 \end{bmatrix}$, $s_1 = \dfrac{h_{21}}{\sqrt{h_{11}^2 + h_{21}^2}}$ and $c_1 = \dfrac{h_{11}}{\sqrt{h_{11}^2 + h_{21}^2}}$.

After m times plane rotation, the following form can be derived as:

$$\boldsymbol{\Omega}_m \cdots \boldsymbol{\Omega}_2 \boldsymbol{\Omega}_1 \bar{\mathbf{H}}_m = \begin{bmatrix} h_{11}^{(m)} & h_{12}^{(m)} & \cdots & h_{1m-1}^{(m)} & h_{1m}^{(m)} \\ & h_{22}^{(m)} & & h_{2m-1}^{(m)} & h_{2m}^{(m)} \\ & & \ddots & & \\ & & & \ddots & \vdots \\ & & & & h_{mm}^{(m)} \\ & & & & 0 \end{bmatrix} \quad (4.13)$$

and

$$\boldsymbol{\Omega}_m \cdots \boldsymbol{\Omega}_2 \boldsymbol{\Omega}_1 \beta \mathbf{e}_1^{(m+1)} = \begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \vdots \\ \gamma_{m+1} \end{bmatrix}. \tag{4.14}$$

By defining $\mathbf{Q}_m \equiv \boldsymbol{\Omega}_m \cdots \boldsymbol{\Omega}_2 \boldsymbol{\Omega}_1 \in \mathbf{R}^{m+1,m+1}$, the Eq.(4.13) and (4.14) becomes:

$$\bar{\mathbf{R}}_m = \mathbf{Q}_m \bar{\mathbf{H}}_m \tag{4.15}$$

$$\bar{\mathbf{g}}_m = \mathbf{Q}_m \beta \mathbf{e}_1^{(m+1)} = \left(\gamma_1, ..., \gamma_{m+1}\right)^T. \tag{4.16}$$

In the end, the final form of the Eq.(4.10) can be rewritten by Eqs.(4.15) and (4.16) as :

$$J(\mathbf{y}_m) = \min \left\| \left(\bar{\mathbf{g}}_m - \bar{\mathbf{R}}_m \mathbf{y}_m \right) \right\|. \tag{4.17}$$

The solution of the above least-square problems can be done by simply solving the triangular system $\bar{\mathbf{R}}_m$. In addition, the linear system is overdetermined, because $\bar{\mathbf{R}}_m \in R^{m+1,m}$, $\mathbf{y}_m \in R^{m,1}$ and, $\bar{\mathbf{g}}_m \in R^{m+1,1}$. Thus, the last element of $\bar{\mathbf{g}}_m$ is the solution of the least square[57], which means:

$$\gamma_{m+1} = \left\| \left(\bar{\mathbf{g}}_m - \bar{\mathbf{R}}_m \mathbf{y}_m^* \right) \right\| \quad \text{where } \mathbf{y}_m^* \text{: solution vector.} \tag{4.18}$$

If the number of basis created by Arnoldi process is same as *n*, which is the dimension of the original matrix **A**, the algorithm is terminated with having the solution $\mathbf{x}_n = \mathbf{x}_0 + \mathbf{V}_n \mathbf{y}_n$. This is called by the Full Orthogonalization Method (FOM). However, the FOM can be too expensive in terms of memory and computational resources, as *n* is increased. The restrated GMRES, instead, creates the set of orthonormal vectors until a certain number *m*, and repeats the loop until the residual meets the convergence condition. Table 4-5 shows the restarted GMRES algorithm.

Table 4-5. Restarted GMRES algorithm

Choose $\mathbf{x}_0$ and compute $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$ and $\mathbf{v}_1 = \frac{\mathbf{r}_0}{\|\mathbf{r}_0\|}$

% Arnoldi Process in Krylov Subspace

For *j*=1:m

$\mathbf{h}_{ij} = \left\langle \mathbf{A}\mathbf{v}_j, \mathbf{v}_i \right\rangle$ for *i*=1,2….,*j*

$\tilde{\mathbf{v}}_{j+1} = \mathbf{A}\mathbf{v}_j - \sum_{i=1}^{j} h_{ij}\mathbf{v}_i$

$h_{j+1,j} = \left\|\tilde{\mathbf{v}}_{j+1}\right\|$

$\mathbf{v}_{j+1} = \tilde{\mathbf{v}}_{j+1} / h_{j+1,j}$

End

$\mathbf{x}_m = \mathbf{x}_0 + \mathbf{V}_m\mathbf{y}_m$ *where* $\mathbf{y}_m$ *minimize* $J(\mathbf{y}_m)$

If $\mathbf{r}_m = \mathbf{b} - \mathbf{A}\mathbf{x}_m$ satisfies convergence condition then stop

Else compute $\mathbf{x}_0 := \mathbf{x}_m$, $\mathbf{v}_1 = \mathbf{r}_m / \|\mathbf{r}_m\|$ and repeat the loop

4.2.3. Preconditioning

The convergence rate of the KSMs is related with the condition number of matrix, which is defined as:

$$R(\kappa) \approx \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1} \text{ where } \kappa(\mathbf{A}) = \|\mathbf{A}\|\|\mathbf{A}^{-1}\|. \tag{4.19}$$

The faster convergence rate is possible, as the condition number of the matrix is close to 1.

The preconditioning is a scheme to accelerate the convergence behaviors of the KSMs. The purpose of the preconditioning is to modify a matrix to have a low condition number, but keep it producing same solutions. There are several types of preconditioners, but the right forward type is applied in this work as:

$$\mathbf{A}\mathbf{P}^{-1}\mathbf{y} = \mathbf{b} \text{ and } \mathbf{x} = \mathbf{P}^{-1}\mathbf{y} \tag{4.20}$$

where **P** is a preconditioner matrix. The algorithms of the preconditioned BiCGSTAB and GMRES are written at Table 4-6 and Table 4-7 respectively. Those algorithms have small modifications from the original algorithms by involving linear systems with a preconditioner matrix.

In order to do efficient preconditioning, a preconditioner matrix has to have following characteristics: 1) the linear system including the preconditioner matrix can be easily solved with small computational load, and 2) The characteristic of the

preconditioner matrix is necessary to remain as similar as that of the original matrix. The first condition is related with the additional floating-point operations (FLOPs) by adopting preconditioning, and the second condition represents how much convergence rate is improved by the reduced condition number. Those racing factors should be considered, when one selects a preconditioner.

It was shown that the Blockwise Incomplete LU factorization (BILU)[28] preconditioned BiCGSTAB could show a considerable execution time reduction in solving the linearized continuity equation, when it compared with conventional iterative methods[23]. To apply the BILU preconditioner in the 3D domains, an additional treatment is necessary to handle outermost matrix elements, because the original BILU was formulated for two-dimensional problems. The BILU3D preconditioner uses Symmetric Gauss-Seidel (SGS) factorization for outermost 1D elements and applies the BILU factorization for inner 2D elements[62]. In following sections, the derivation of BIL3D preconditioner[30] will be described.

Table 4-6. Preconditioned BiCGSTAB algorithm

$\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$ for an initial guess $\mathbf{x}_0$;
$\rho_0 = \alpha = \omega_0 = 1$; $\mathbf{v}_0 = \mathbf{p}_0 = 0$
For *i*=1,2,3…
$\rho_i = (\mathbf{r}_0, \mathbf{r}_{i-1})$; $\beta = (\rho_i / \rho_{i-1})(\alpha / \omega_{i-1})$; $\mathbf{p}_i = \mathbf{r}_{i-1} + \beta(\mathbf{p}_{i-1} - \omega_{i-1} v_{i-1})$
Solve $\mathbf{P}\mathbf{y} = \mathbf{p}_i$ for $\mathbf{y}$
$\mathbf{v}_i = \mathbf{A}\mathbf{y}$
$\alpha = \rho_i / (\mathbf{r}_0, \mathbf{v}_i)$; $\mathbf{s} = \mathbf{r}_{i-1} - \alpha \mathbf{v}_i$
Solve $\mathbf{P}\mathbf{z} = \mathbf{s}$ for $\mathbf{z}$
$\mathbf{t} = \mathbf{A}\mathbf{z}$
$\omega_i = (\mathbf{t}, \mathbf{s}) / (\mathbf{t}, \mathbf{t})$; $\mathbf{x}_i = \mathbf{x}_{i-1} + \alpha \mathbf{y} + \omega_i \mathbf{z}$
If $\mathbf{x}_i$ is converged, then quit
$\mathbf{r}_i = \mathbf{s} - \omega_i \mathbf{t}$
End

Table 4-7. Preconditioned GMRES algorithm

Choose $\mathbf{x}_0$ and compute $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}$ and $\mathbf{v}_1 = \dfrac{\mathbf{r}_0}{\|\mathbf{r}_0\|}$

% Arnoldi Process in Krylov Subspace

For *j*=1:m

Solve $\mathbf{M}\mathbf{z}_j = \mathbf{v}_j$ where **M** is preconditioner of **A**

$\mathbf{w}_j = \mathbf{A}\mathbf{z}_j$

$h_{ij} = \langle \mathbf{w}_j, \mathbf{v}_i \rangle$ for *i*=1,2….,*j*

$\tilde{\mathbf{v}}_{j+1} = \mathbf{w}_j - \sum_{i=1}^{j} h_{ij}\mathbf{v}_i$

$h_{j+1,j} = \|\tilde{\mathbf{v}}_{j+1}\|$

$\mathbf{v}_{j+1} = \tilde{\mathbf{v}}_{j+1} / h_{j+1,j}$

End

$\mathbf{x}_m = \mathbf{x}_0 + \mathbf{Z}_m\mathbf{y}_m$ *where* $\mathbf{y}_m$ *minimize* $J(\mathbf{y}_m)$ and $\mathbf{Z}_m = [\mathbf{z}_1, ..., \mathbf{z}_m]$

If $r_m = b - A_m$ satisfies convergence condition then stop

Else compute $x_0 := x_m$, $v_1 = r_m / \|r_m\|$ and repeat the loop

(1) Symmetric Gauss-Seidel Factorization

The recursion relation of the LU factorization for a block tridiagonal matrix **A** can be formulated as:

$$\Delta_1 = \mathbf{D}_1$$

$$\Delta_m = \mathbf{D}_m - \mathbf{L}_m\Delta_{m-1}^{-1}\mathbf{U}_{m-1}, \qquad m = 2, ..., n$$

$$\mathbf{A} = (\mathbf{L} + \Delta)\Delta^{-1}(\Delta + \mathbf{U}). \tag{4.21}$$

where **L** : strictly lower triangular parts of **A**

U : strictly upper triangular parts of **A**

Δ : a block diagonal matrix consisting of blocks of Δ_m.

Subscript *m* denotes the index of diagonal blocks and *n* is the number of blocks in the diagonal. The complete LU factorization involves computational loads because Δ_m becomes a full matrix by the inverse of Δ_{m-1}. This can harm the efficiency of a preconditioner. The Symmetric Gauss-Seidel (SGS) factorization avoids the associated problem by simply taking the diagonal matrices as the submatrices as below:

$$\Delta_m \approx \mathbf{D}_m . \tag{4.22}$$

The SGS factorized matrix $\mathbf{P}_{GS}$ has difference with the original matrix as:

$$\begin{aligned}\mathbf{P}_{GS} &= (\mathbf{L}+\mathbf{D})\mathbf{D}^{-1}(\mathbf{D}+\mathbf{U}) = \mathbf{L}+\mathbf{D}+\mathbf{U}+\mathbf{L}\mathbf{D}^{-1}\mathbf{U} \\ &= \mathbf{A}+\mathbf{R}.\end{aligned} \tag{4.23}$$

This means that the SGS factorization can be a good approximation when the matrix **R** is small. In the neutronics code, the heterogeneity of the radial direction is large, whereas that of the axial direction is small. Hence, the SGS factorization is applied to the axial coupling. However, the elements in the pressure matrix might have strong coupling to the axial direction rather than the radial direction by following reasons. Firstly, flows in most problems are axially dominant, so the velocities of axial flows are much larger than that of crossflows. Secondly, the pressure gradients also are higher in the axial direction than in the radial direction. The axial flows undergo more pressure losses by gravitational acceleration and friction loss. Moreover, the deviation of densities is relatively large in the axial direction, when heats are provided by rods.

In those reasons, the SGS factorization in ESCOT is applied to the y-directional coupling, and it requires to order the elements of the linear system involving the pressure equation properly[30]. The linear system involving the pressure equation (Eq.(2.88)) can be written as:

$$\mathbf{Ap}=\mathbf{s} \tag{4.24}$$

where

Pressure matrix: $\mathbf{A} \equiv \mathbf{A}^{(3)} \in \mathbf{R}^{N,N}$

Solution column vector: $\mathbf{p} = col(p_1, \cdots, p_N) \in \mathbf{R}^N$,

Source column vector: $\mathbf{s} = col(s_1, \cdots, s_N) \in \mathbf{R}^N$

and N is the number of total meshes in a solution domain. In addition, the submatrix $\mathrm{A}^{(d)}$ representing the coefficient matrix in d-dimension is defined as:

$$\mathbf{A}^{(d)} \equiv \begin{bmatrix} \mathbf{A}_1^{(d-1)} & \mathbf{U}_1^{(d)} & & & & & \\ \mathbf{L}_2^{(d)} & \mathbf{A}_2^{(d-1)} & \mathbf{U}_2^{(d)} & & & & \\ & \cdot & \cdot & \cdot & & & \\ & & \cdot & \cdot & \cdot & & \\ & & & \mathbf{L}_{n-1}^{(d)} & \mathbf{A}_{n-1}^{(d-1)} & \mathbf{U}_{n-1}^{(d)} & \\ & & & & \mathbf{L}_n^{(d)} & \mathbf{A}_n^{(d-1)} \end{bmatrix} \tag{4.25}$$

where n is the number of planes (NZ), rows (NY), or columns (NX) corresponding to the value of d. The matrix $\mathbf{A}_l^{(d)}$ is a block tridiagonal matrix or tridiagonal matrix, whereas $\mathbf{L}_l^{(d)}$ and $\mathbf{U}_l^{(d)}$ are diagonal matrices defined as:

$$\left.\begin{aligned}\mathbf{L}_l^{(d)} &\equiv diag(l_{p+1}^w,\cdots,l_{p+q}^w)\\ \mathbf{U}_l^{(d)} &\equiv diag(u_{p+1}^w,\cdots,u_{p+q}^w)\end{aligned}\right\}\in\mathbf{R}^{q,q} \tag{4.26}$$

where the elements consist of coefficients of neighbors given in Eq.(2.88), and *w* is one of the coupling directions (*x*,*y*,*z*). Also, *p* and *q* are properly defined by corresponding blocks. The ordering of linear system depends on how *w* is linked with the value of *d*.

The ordinary way to order of elements in a linear system dealing with 3-D domain problems is that columns (x-direction) are first, and rows (y-direction) are second, and planes (z-direction) are last. This implies that *w*=*x*, *y*, and *z* for *d*=1, 2, and 3 respectively by the above definitions. The elements in the linear system shown in Figure 2-5 are filled by the ordinary ordering. However, to apply SGS factorization to the y-directional coupling, the ordering of elements should be changed into *w*=*z*, *x*, and *y* for *d*=1, 2, and 3 respectively. In other words, the elements are filled by the order of planes firstly, columns secondly, and rows lastly.

Thus, the preconditioner matrix can be factorized by the SGS factorization as below:

$$\mathbf{P}_{GS}=\left(\mathbf{L}^{(3)}+\mathbf{D}^{(3)}\right)\left(\mathbf{I}+\mathbf{D}^{(3)^{-1}}\mathbf{U}^{(3)}\right)=\hat{\mathbf{L}}_{GS}^{(3)}\hat{\mathbf{U}}_{GS}^{(3)} \tag{4.27}$$

where $\hat{\mathbf{L}}_{GS}^{(3)}\equiv\begin{bmatrix}\mathbf{A}_1^{(2)} & & & & \\ \mathbf{L}_2^{(3)} & \mathbf{A}_2^{(2)} & & & \\ & \ddots & \ddots & & \\ & & \ddots & \ddots & \\ & & & \mathbf{L}_{\mathrm{NY}}^{(3)} & \mathbf{A}_{\mathrm{NY}}^{(2)}\end{bmatrix}$ and

$$\hat{\mathbf{U}}_{GS}^{(3)}\equiv\begin{bmatrix}\mathbf{I} & \mathbf{A}_1^{(2)^{-1}}\mathbf{U}_1^{(3)} & & & \\ & \mathbf{I} & \ddots & & \\ & & \ddots & \ddots & \\ & & & \ddots & \mathbf{A}_{\mathrm{NY}-1}^{(2)^{-1}}\mathbf{U}_{\mathrm{NY}-1}^{(3)} \\ & & & & \mathbf{I}\end{bmatrix}.$$

Even though the pressure matrix is factorized by the method, the diagonal block matrices $\mathbf{A}_j^{(2)}$ are penta-diagonal matrices that are not easy to solve with the direct method. Hence, the BILU factorization is applied to $\mathbf{A}_j^{(2)}$

.

(2) BILU factorization

The idea of the BILU factorization is to keep a sparsity of submatrices as same as that of an original matrix. In other words, the BILU factorization updates non-zero elements in the matrix as the LU factorization does, and leaves zero elements.

Let us consider the factorization of a penta-diagonal matrix $\mathbf{A}^{(2)}$ at a certain row *j* in Eq.(4.27). In the BILU factorization, the recursive relation of Eq.(4.21) is modified to:

$$\Delta_i = \mathbf{A}_i^{(1)} - \mathbf{L}_i^{(2)} \Omega\left(\Delta_{i-1}^{-1}\right) \mathbf{U}_{i-1}^{(2)}, \qquad i = 2,..., NX \tag{4.28}$$

where $\mathbf{A}_i^{(1)}$ is the *NZ*-by-*NZ* tridiagonal matrix at *i*-th column, and $\mathbf{L}_i^{(2)}$ and $\mathbf{U}_i^{(2)}$ are *NZ*-by-*NZ* diagonal matrices consisting of coefficients of neighbors in x-direction at *i*-th column. For Δ_i to become the tridiagonal matrix, $\Omega(\Delta_{i-1}^{-1})$ has to be a tridiagonal matrix as well. Originally, Δ_{i-1} contains elements as below:

$$\begin{aligned}\Delta_{i-1} &= \left(\mathbf{H} + \mathbf{\Gamma}\right) \mathbf{\Gamma}^{-1} \left(\mathbf{\Gamma} + \mathbf{F}\right) \\ &= \begin{bmatrix} g_{1,i-1} & & & \\ h_{2,i-1} & \ddots & & \\ & \ddots & \ddots & \\ & & h_{NZ,i-1} & g_{NZ,i-1} \end{bmatrix} \begin{bmatrix} g_{1,i-1}^{-1} & & & \\ & \ddots & & \\ & & \ddots & \\ & & & g_{NZ,i-1}^{-1} \end{bmatrix} \begin{bmatrix} g_{1,i-1} & f_{1,i-1} & & \\ & \ddots & \ddots & \\ & & \ddots & f_{NZ-1,i-1} \\ & & & g_{NZ,i-1} \end{bmatrix}\end{aligned} \tag{4.29}$$

where $h_{k,i} = \mathbf{L}_{k,i}^{(1)} - \mathbf{L}_{k,i}^{(2)} \left[\Omega\left(\Delta_{i-1}^{-1}\right)\right]_{k,k-1} \mathbf{U}_{k-1,i-1}^{(2)}$

$f_{k,i} = \mathbf{U}_{k,i}^{(1)} - \mathbf{L}_{k,i}^{(2)} \left[\Omega\left(\Delta_{i-1}^{-1}\right)\right]_{k,k+1} \mathbf{U}_{k+1,i-1}^{(2)}$

$g_{k,i} = \mathbf{G}_{k,i} - h_{k,i} g_{k-1,i}^{-1} f_{k-1,i}$

$\mathbf{G}_{k,i} = \mathbf{A}_{k,i}^{(0)} - \mathbf{L}_{k,i}^{(2)} \left[\Omega\left(\Delta_{i-1}^{-1}\right)\right]_{k,k} \mathbf{U}_{k,i-1}^{(2)}$.

For brevity, let us define Δ_{i-1}^{-1} as **E**. Then, the below relations can be derived from Eq.(4.29) as:

$$\mathbf{E} = \left(\mathbf{H} + \mathbf{\Gamma}\right)^{-1} - \mathbf{\Gamma}^{-1} \mathbf{F} \mathbf{E} \tag{4.30}$$

and

$$\mathbf{E} = \left(\mathbf{\Gamma} + \mathbf{F}\right)^{-1} - \mathbf{E} \mathbf{H} \mathbf{\Gamma}^{-1}. \tag{4.31}$$

The first term and second term in Eq.(4.30) have following matrix elements:

$$\left(\mathbf{H}+\mathbf{\Gamma}\right)^{-1}=\begin{bmatrix} g_1^{-1} & & & \\ \cdot & \cdot & & \\ \cdot & \cdot & \cdot & \\ \cdot & \cdot & \cdot & g_{NZ}^{-1} \end{bmatrix} \tag{4.32}$$

and

$$\mathbf{\Gamma}^{-1}\mathbf{F}\mathbf{E}=\begin{bmatrix} g_1^{-1}f_1e_{21} & g_1^{-1}f_1e_{22} & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & g_{NZ-1}^{-1}f_{NZ-1}e_{NZ,NZ-1} & g_{NZ-1}^{-1}f_{NZ-1}e_{NZ,NZ} \\ 0 & 0 & \cdot & 0 \end{bmatrix}. \tag{4.33}$$

Therefore, following recursive relations for diagonal and upper diagonal elements of **E** can be derived as:

$$\begin{aligned} e_{NZ,NZ} &= g_{NZ}^{-1} \\ e_{k-1,k} &= -g_{k-1}^{-1}f_{k-1}e_{k,k} \\ e_{k-1,k-1} &= g_{k-1}^{-1}\left(1-f_{k-1}e_{k,k-1}\right). \end{aligned} \tag{4.34}$$

By using Eq.(4.31), the recursive relation for lower diagonal elements of **E** can be derived as:

$$e_{k,k-1}=-e_{k,k}h_k\,f_{k-1}^{-1} \tag{4.35}$$

Now, $\Omega(\Delta_{i-1}^{-1})$ becomes a tridiagonal matrix by the above recursive relations, and the BILU factorization can be performed by using Eq.(4.28). Then, $\mathbf{A}_j^{(2)}$ can be approximated by its BILU factor $\mathbf{P}_j^{(2)}$ as:

$$\mathbf{P}_j^{(2)}=(\mathbf{L}_j^{(2)}+\Delta_j)(\mathbf{I}+\Delta_j^{-1}\mathbf{U}_j^{(2)})\equiv\hat{\mathbf{L}}_j^{(2)}\hat{\mathbf{U}}_j^{(2)} \tag{4.36}$$

By replacing $\mathbf{A}_j^{(2)}$ with $\mathbf{P}_j^{(2)}$ in Eq.(4.27), the BILU3D preconditioner is formulated as:

$$\mathbf{P}=\hat{\mathbf{L}}\hat{\mathbf{U}}=\begin{bmatrix} \mathbf{P}_1^{(2)} & & & & \\ \mathbf{L}_2^{(3)} & \mathbf{P}_2^{(2)} & & & \\ & \cdot & \cdot & & \\ & & \cdot & \cdot & \\ & & & \mathbf{L}_{\mathrm{NY}}^{(3)} & \mathbf{P}_{\mathrm{NY}}^{(2)} \end{bmatrix}\begin{bmatrix} \mathbf{I} & \mathbf{P}_1^{(2)^{-1}}\mathbf{U}_1^{(3)} & & & \\ & \mathbf{I} & \cdot & & \\ & & \cdot & \cdot & \\ & & & \cdot & \mathbf{P}_{\mathrm{NY}-1}^{(2)^{-1}}\mathbf{U}_{\mathrm{NY}-1}^{(3)} \\ & & & & \mathbf{I} \end{bmatrix}. \tag{4.37}$$

(3) Solution of the preconditioner equation

The solution of the BILU3D preconditioner equation can be obtained by three levels of forward and backward substitutions. Let us assume that **Px**=**b** is the linear system to be solved. The forward substitution of the highest level is:

$$\mathbf{P}_j^{(2)}\mathbf{y}_j = \mathbf{b}_j - \mathbf{L}_j^{(3)}\mathbf{y}_{j-1},\ j = 2...NY \tag{4.38}$$

and that of the backward substitution is:

$$\mathbf{P}_j^{(2)}\boldsymbol{\zeta}_j = \mathbf{U}_j^{(3)}\mathbf{x}_{j+1},\ j = NY-1...1 \tag{4.39}$$

where $\boldsymbol{\zeta}_j = \mathbf{y}_j - \mathbf{x}_j$. The 2nd level of forward and backward substitution is necessary to solve the linear systems in Eqs.(4.38) and (4.39). The 2nd level of forward substitution is:

$$\Delta_{i,j}^{(2)}\mathbf{y}_{i,j} = \hat{\mathbf{b}}_{i,j} - \mathbf{L}_j^{(2)}\mathbf{y}_{i-1,j},\ i = 2...NX \tag{4.40}$$

where $\hat{\mathbf{b}}_{i,j} = \mathbf{b}_{i,j} - \mathbf{L}_{i,j}^{(3)}\mathbf{y}_{i,j-1}$.The linear system involving tridiagonal matrix $\Delta_{i,j}^{(2)}$ can be solved by the last level of forward and backward substitution. The last level of forward substitution can be formulated as:

$$g_{k,i,j}y_{k,i,j} = \tilde{b}_{k,i,j} - h_{k,i,j}y_{k-1,i,j},\ k = 2...NZ \tag{4.41}$$

where $\tilde{b}_{k,i,j} = \hat{b}_{k,i,j} - L_{k,i,j}^{(2)}y_{k,i-1,j}$. The equations for 2nd and last level of backward substitution are not written in here, because they can be easily formulated by the similar way with Eq.(4.39). By those hierarchical forward and backward substitutions, the solutions of the preconditioner equation can be obtained.

4.3. Performance of the Krylov Subspace Method

In order to find the most effective linear solver to solve linear equations in the SIMPLE algorithm, the performance test of various linear solvers was carried out with two sample problems. The sample problems were design to have single-phase flows under the normal operation condition of PWRs. Five linear solvers including one direct solver and four KSMs stood as candidates; They are SuperLU, unpreconditioned GMRES/BiCGSTAB, and BILU preconditioned GMRES /BiCGSTAB. The performance of linear solvers is measured by comparing with that of SuperLU as reference. The result is discussed in this section.

4.3.1. 1x1 assembly test problem

The same 1x1 assembly problem in section 4.1 is used for the performance test of the KSMs. Table 4-8 shows the test result in terms of computing time and the number of inner iterations, and the schematic graph about computing time is illustrated at Figure 4-1.

Firstly, let us compare the result of unpreconditioned KSMs with that of SuperLU. It can be figured out that the computing time for solving momentum equations is significantly reduced, when the unpreconditioned KSMs are applied. The speedup with a factor of 65 and 390 is achieved by GMRES and BiCGSTAB respectively. For solving the pressure equation, however, only the BiCGSTAB shows a time reduction by a factor of 21. GMRES takes even more time than SuperLU, although the number of inner iterations of GMRES is less than that of BiCGSTAB. That is because the FLOPs of GMRES at one inner iteration is much higher than that of BiCGSTAB.

Secondly, let us compare the result of BILU preconditioned KSMs with that of unpreconditioned KSMs. Note that the BILU preconditioning is applied only for solving the pressure equation. By introducing the preconditioner, the GMRES achieves a great computing time reduction by a factor of 5.3, while the computing time of BiCGSTAB is rather slightly increased. The poor performance of the BILU preconditioner in BiCGSTAB can be interpreted in the aspect of inner iteration. By adopting the BILU preconditioner, the number of inner iteration is decreased in both KSMs as expected. However, the scale of reduction is different by 26 times and 6

times in case of GMRES and BiCGSTAB respectively. The ineffective reduction of inner iteration causes the poor performance of BILU preconditioned BiCGSTAB.

Lastly, it can be concluded by the result that the unpreconditioned BiCGSTAB is the most effective linear solvers among candidates. Nevertheless, GMRES can be suggested as a secondary linear system solver in ESCOT due to its stability, even though the computing time is much slower than BiCGSTAB. The stability of linear solvers is verified by the behavior of relative residual norms during inner iterations. The relative residual norm is defined as:

$$\text{Relative residual norm:} = \frac{\left\| \mathbf{r}_{i,j} \right\|}{\left\| \mathbf{b}_{0,j} \right\|} = \frac{\left\| \mathbf{b}_{0,j} - \mathbf{A}\mathbf{x}_{i,j} \right\|}{\left\| \mathbf{b}_{0,j} \right\|}. \tag{4.42}$$

where the subscript *i* indicates the index of inner iteration and *j* denotes the index of outer iteration. In Figure 4-2, the relative residual norm of BiCGSTAB and GMRES is drawn up to the 3rd outer iteration. While the behavior of BiCGSTAB shows some peaks during inner iteration, the behavior of GMRES is monotonously decreasing shape. By this characteristics of GMRES, it guarantees a better numerical stability than BiCGSTAB.

Table 4-8. Computing time of 1x1 assembly test problem with various linear solvers

	SuperLU	**Unpreconditioned KSMs**		**BILU preconditioned KSMs**	
		GMRES	BiCGSTAB	GMRES	BiCGSTAB
Solving pressure Eq. (sec)	159.3	168.8	7.4	25.8	9.2
Solving momentum Eq. (sec)	156.3	2.4	0.4	2.4	0.4
Total CPU time (sec)	324.7	175.8	12.3	32.9	14.2
Inner iteration for pressure Eq.	-	23,241	32,576	891	5,332

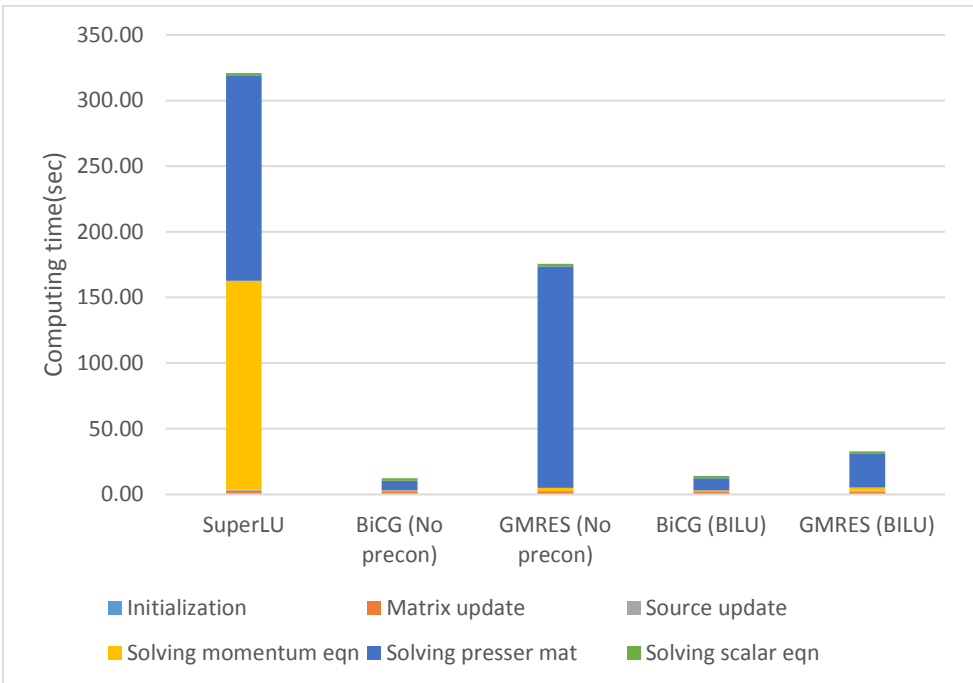

Figure 4-1. Computing time of 1x1 assembly test problem with various linear solvers

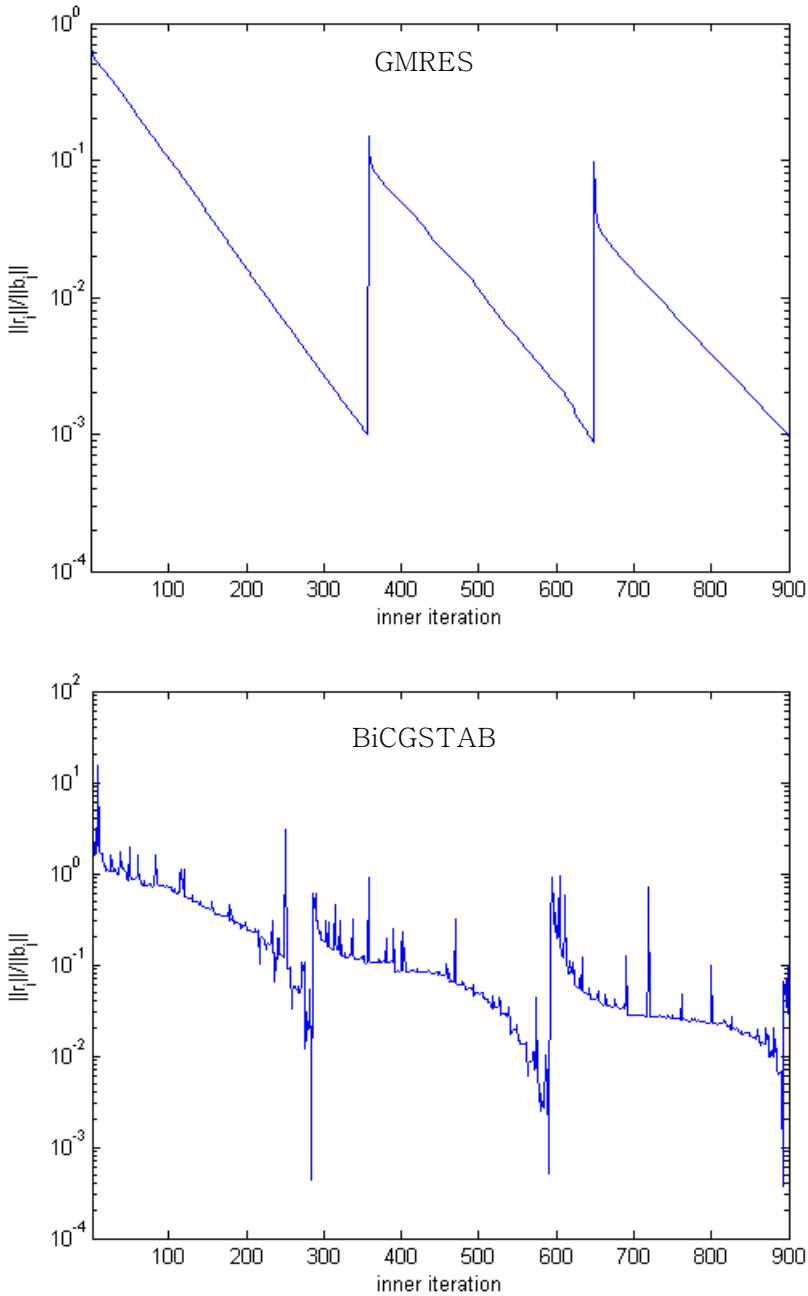

Figure 4-2. Behavior of the relative residual norm change of GMRES and BiCGSTAB in the 1x1 assembly problem

4.3.2. 2x2 assembly with checkerboard-shape power profile

The second test was designed to have a 2x2 array of fuel assemblies with checkerboard-shape power profile. Unlike the previous test in section 4.3.1, the problem has radial power distribution by giving 1.2 and 0.8 nominal power in the checkerboard shape. The number of total meshes is 43,560 (33×33×40). The temperature profile calculated by ESCOT is given at Figure 4-3.

Table 4-9 shows the result of its performance test. The tendency of the result is similar as that of single assembly problem, but some points are different.

When the comparison between SuperLU and unpreconditioned KSMs is concerned, both KSMs show the improved performance than SuperLU. In case of computing time for solving momentum equations, the time is reduced by a factor of 403 (GMRES) and 2751 (BiCGSTAB). Moreover, unlike the result of the single assembly problem, the computing time for solving the pressure equation is decreased in both KSMs by a factor of 4.8 (GMRES) and 117 (BiCGSTAB). From the comparison, the following two conclusions can be drawn. Firstly, the KSMs are more efficient on solving linear systems involving momentum equations than pressure equation. Secondly, better efficiency of the KSMs can be achieved when larger size problems are dealt with.

As like in the single assembly problem, the BILU preconditioning could improve the performance of GMRES only. GMRES could obtain speedup by a factor of 3.5, while BiCGSTAB slowed down by a factor of 1.8. One noticeable thing is that the efficiency of the BILU preconditioning is lowered, when it compared with the result of the single assembly problem. The reduction scale of inner iterations is a factor of 13.4 (GMRES) and 3.8 (BiCGSTAB), which is half of that in the single assembly problem. The lowered performance can be explained by the assumption applied to BILU3D derivation. In the derivation, the Symmetric Gauss-Seidel (SGS) factorization is applied to matrix elements which are neighboring in y-direction. And SGS factorization can be a good approximation only if neighboring elements have a week coupling. In this problem, the coupling effect in y-direction is stronger than that in the previous problem by the asymmetric power profile and corresponding energy dependent variables. Thus, it can be concluded that applying BILU3D with the current linear system involving the pressure equation deteriorates the solution performance for the non-uniform power distribution cases.

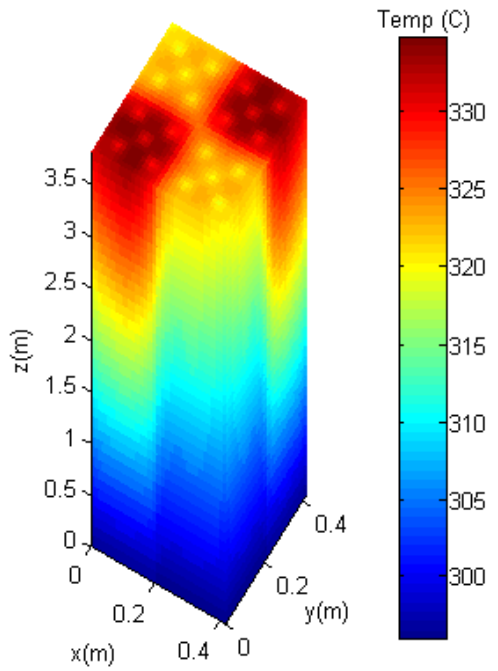

Figure 4-3. Temperature profile of 2x2 checkerboard problem

Table 4-9. Computing time of 2x2 checkerboard problem with various linear solvers

	SuperLU	**Unpreconditioned KSMs**		**BILU preconditioned KSMs**	
		GMRES	BiCGSTAB	GMRES	BiCGSTAB
Solving pressure Eq. (sec)	3,639	750.2	31.5	215.6	57.2
Solving momentum Eq. (sec)	4,678	11.6	1.7	11.6	1.7
Total CPU time (sec)	8,356	61.0	789.5	255.1	87.0
Inner iteration for pressure Eq.	-	22,803	32,986	1,693	8,527

Chapter 5. Parallelization

In order to be capable of analyzing the full-core-size problems, the parallel execution of the code is essential. As multi-core systems are mounted on many computing machines, heavy calculations can be split and shared by processors. OpenMP is one of the most famous parallel programming tools based on shared memory. Not only by shared memory, but the task can be distributed between machines by the mutual communication. This is the massage-passing system or the distributed memory system. Message Passing Interface (MPI) is a good example. To fully utilize multi-cluster, multi-core parallel machines, MPI is better choice than OpenMP. However, OpenMP is selected in this work, because it can easily parallelize codes with slight modifications of code scripts. It is the choice for the initial version of the code and the code will be parallelized with MPI as well in the future research.

The best way to parallelize a code is to parallelize whole parts of the code, but it is impossible because some parts should be performed in serial. For the efficient parallelization, the most time-consuming parts should be parallelized in the first priority. As shown in Chapter 4, solving the pressure equation is the most time-consuming part. It takes 60% over total computing time in the single assembly problem. Therefore, the BiCGSTAB and GMRES are parallelized to reduce the computing time to solve linear systems.

The algorithms of the Krylov subspace methods consist of many multiplying operations between vectors and matrices. Those operations are naturally parallelizable, so they can be easily parallelized. However, in case of the preconditioned Krylov subspace methods, the forward and backward substitutions are involved, which has to be performed in serial but those are time-consuming operations. To resolve this problem, a domain decomposition with some approximations is necessary. Two methods for the domain decomposition are explained in the following section.

5.1. Incomplete Domain Decomposition Preconditioning

The linear system including a preconditioner can be written as:

$$\mathbf{Pz} = \mathbf{r} \tag{5.1}$$

where **P** is a preconditioner matrix. The preconditioner matrix can be factorized into multiplication of a lower triangular matrix **L** and an upper triangular matrix **U** as follows:

$$\mathbf{LUz} = \mathbf{r} \tag{5.2}$$

and a linear system for forward substitution can be written as:

$$\mathbf{Ly} = \mathbf{r}\,. \tag{5.3}$$

where **y**=**Uz**.

If a domain decomposition with two subdomains is assumed, the linear system in Eq.(5.3) can be partitioned into block matrices and block vectors as follows:

$$\begin{bmatrix} \mathbf{L}_{11} & \mathbf{0} \\ \mathbf{L}_{21} & \mathbf{L}_{22} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \end{bmatrix} \tag{5.4}$$

where subscripts imply domain number. Solving the region 1 can be independently done, because it does not have any coupling with the information of the region 2. However, when the solution in the region 2 is concerned, it cannot be solved until $\mathbf{y}_1$ is given.

The first approximation method is to completely ignore the coupling belonging to neighboring domains[29]. In the above example, this approximation means as:

$$\mathbf{y}_1 \approx 0\,. \tag{5.5}$$

The method is easy to be implemented in a code, and it does not require additional FLOPs. However, the increment of the number of iterations until convergence is expected to be considerable due to its rough approximation, and it harms parallel efficiency. For example, the ideal speedup with 2 processors will be a factor of two. However, if the number of iterations is increased 50% by applying the approximation, then the maximum achievable speedup will be a factor of 1.5. This can be more severe when the neighboring effect between domains is stronger.

The second method is to approximate the solutions of neighboring with diagonal elements only[30], and it can be expressed in the example as follows:

$$\mathbf{y}_1 = \mathbf{L}_{11}^{-1}\mathbf{r}_1 \approx diag(\mathbf{L}_{11})^{-1}\mathbf{r}_1\,. \tag{5.6}$$

In this case, the additional FLOPs are required to obtain the approximated solutions belonging to other subdomains, but the required FLOPs are affordable. Moreover, applying this method is expected that the increment of the number of iterations will be less than that of the completely ignored one.

In order to figure out which parallelization scheme comes up with the best performance, all of listed parallelization schemes are implemented into ESCOT and their performance are examined with respect to speedup, computing time and efficiency. The results of parallel performance tests are presented in the following section.

5.2. Efficiency of the Parallelization

5.2.1. 2x2 assembly with uniform power distribution

A test problem was set to have a dimension of 2x2 array of fuel assemblies and the uniform power distribution. Unlike the 2x2 checkerboard problem in section 4.3.2, this problem is just a geometrically expanded problem from the single assembly problem. The radial power profile was set to be uniform, because it is turned out that applying BILU3D with the current linear system involving the pressure equation deteriorates the solution performance for the non-uniform power distribution cases. Figure 5-1 illustrates the temperature profile of this sample problem.

The performance of the parallelization schemes applying to the BiCGSTAB and GMRES is compared with three parameters: wall-clock time (computing time), speedup, and efficiency. The parallel efficiency is calculated by the relation as follows:

$$efficiency(\%) = \frac{Speedup}{\#of\ processors} \times 100 \tag{5.7}$$

Three different parallelization schemes were examined. The first one is for the parallelization of the KSMs without preconditioner. In this case, the targets to be parallelized are the operations of matrix-vector multiplication and vector-vector multiplication in KSMs' algorithm. Two other schemes are the Incomplete Domain Decompositions (IDD) for the parallelization of the KSMs with the BILU preconditioner. For brevity, the IDD which neglects neighboring coupling is named as "IDD1" and the IDD which uses the approximation with diagonal terms is named as "IDD2" respectively.

The speedup and efficiency at Table 5-1 and Table 5-2 are calculated with the reference of computing time for solving pressure equation with a single processor. The ideal speedup and speedup of all cases by the number of processes are drawn at Figure 5-2. Firstly, let us compare the results between the unpreconditioned KSMs and preconditioned KSMs. It is figured out that the speedup of the unpreconditioned KSMs is greater than that of the preconditioned KSMs. That is because the unpreconditioned KSMs consist of only the multiplication operations of matrix-vector and vector-vector which are naturally parallelizable. Thus, any approximation by domain decomposition is not necessary, so the number of inner iterations stays in constant regardless of the number of processors used.

Secondly, when the results of each preconditioned KSM with different IDD scheme are compared, the IDD2 always shows the better parallel performance than IDD1. It is closely related with the increment scale of the inner iterations as shown in Table 5-3. The increment scale of inner iterations of IDD2 is always less than that of IDD1. Moreover, the difference becomes larger when the more processors are used for the parallel execution. Therefore, it can be concluded that the diagonal approximation in IDD2 can impede the loss of information at coupled domains in a certain amount. In addition, it can be more effective when the loss of information becomes larger by the increase of the number of processors.

Lastly, it is turned out that the unpreconditioned BiCGSTAB shows the best performance. The best performance of the unpreconditioned BiCGSTAB results from the superior performance in serial calculations and naturally parallelizable operations.

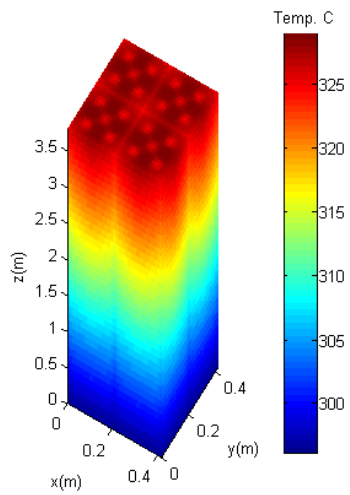

Figure 5-1. Temperature profile of 2x2 uniform power test problem

Table 5-1. Results of parallel 2x2 uniform power problem (1/2)

# of processors	BiCGSTAB				GMRES			
	No preconditioner							
	Total CPU time (sec)	CPU time of pressure Eqn. (sec)	Speedup	Efficiency (%)	Total CPU time (sec)	CPU time of pressure Eqn. (sec)	Speedup	Efficiency (%)
1	64.4	30.9	1.00	100.0	923.3	889.9	1.00	100.0
2	46.5	16.3	1.89	94.5	516.0	486.3	1.83	91.5
4	39.3	9.2	3.36	83.9	300.3	270.6	3.29	82.2
6	37.9	7.1	4.36	72.6	248.1	217.6	4.09	68.2
8	37.7	6.4	4.84	60.5	220.0	189.2	4.70	58.8
10	37.5	5.8	5.33	53.3	216.0	184.5	4.82	48.2
12	37.9	5.5	5.58	46.5	198.4	166.4	5.35	44.6
# of processors	**BiCGSTAB**							
	BILU_IDD1				**BILU_IDD2**			
1	67.7	34.1	1.00	100.0	67.7	34.1	1.00	100.0
2	55.7	25.4	1.35	67.3	53.3	23.0	1.48	74.1
4	45.9	15.8	2.16	54.1	43.2	13.3	2.58	64.4
6	43.4	12.4	2.76	46.1	41.7	10.7	3.19	53.1
8	42.6	11.3	3.02	37.8	40.8	9.4	3.62	45.2
10	41.8	10.2	3.35	33.5	40.3	9.1	3.77	37.7
12	41.2	8.8	3.88	32.3	40.3	8.0	4.29	35.8

Table 5-2. Results of parallel 2x2 uniform power problem (2/2)

# of processors	GMRES							
	BILU_IDD1				BILU_IDD2			
	Total CPU time (sec)	CPU time of pressure Eqn. (sec)	speedup	Efficiency (%)	Total CPU time (sec)	CPU time of pressure Eqn. (sec)	speedup	Efficiency (%)
1	136.2	102.5	1.00	100.0	136.2	102.5	1.00	100.0
2	129.6	99.5	1.03	51.5	115.6	85.6	1.20	59.9
4	97.5	67.5	1.52	38.0	83.7	53.7	1.91	47.7
6	88.2	57.7	1.78	29.6	77.1	46.4	2.21	36.8
8	87.8	56.6	1.81	22.6	79.3	48.0	2.13	26.7
10	94.7	62.9	1.63	16.3	83.7	52.0	1.97	19.7
12	90.2	52.9	1.94	16.1	73.5	41.4	2.48	20.6

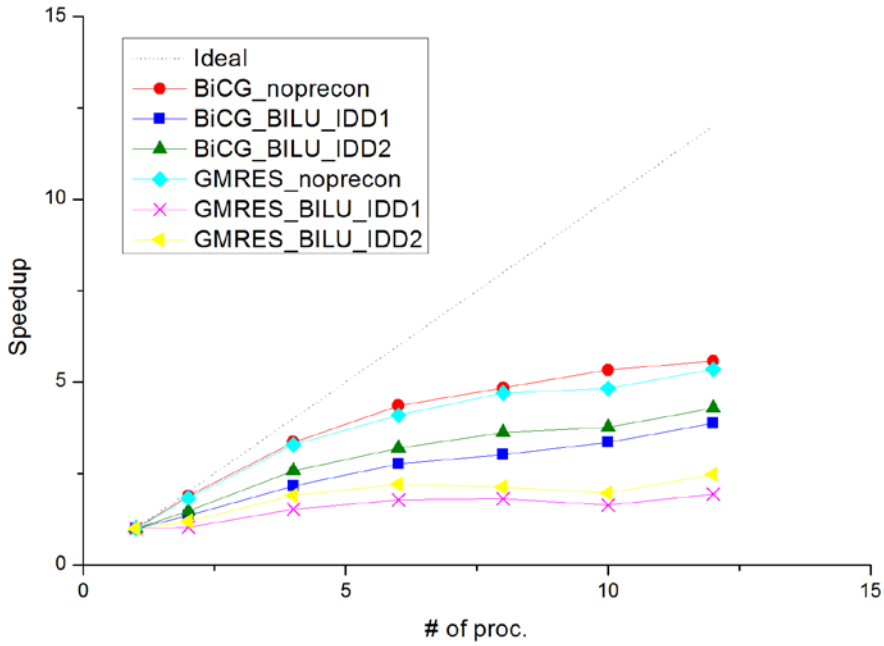

Figure 5-2. Speedup of parallel 2x2 uniform power problem

Table 5-3. The number of inner iterations of parallel 2x2 uniform power problem

	Unpreco-nditioned	**BILU**	**# of proc.**	**BILU _IDD1**		**BILU_IDD2**	
				Inner	Increment scale	Inner	Increment scale
BiCGS-TAB	32,418	5,076	2	7,449	1.47	6,674	1.31
			4	8,847	1.74	7,272	1.43
			6	9,699	1.91	7,846	1.55
			8	10,670	2.10	8,578	1.69
			10	11,757	2.32	9,362	1.84
			12	12,705	2.50	10,067	1.98
GMRES	23,462	749	2	1,547	2.07	1,312	1.75
			4	1,965	2.62	1,481	1.98
			6	2,279	3.04	1,654	2.21
			8	2,625	3.5	1,991	2.66
			10	3,314	4.42	2,378	3.17
			12	3,120	4.17	2,222	2.97

5.2.2. 9x9 mini MSLB problem

Because parallel performances are better at problems with heavy computational works, the bigger size problem is set. The 9x9 MSLB problem which was described in the section 3.3.3 is used. The number of meshes in the problem is 841,000, which is 20 times bigger than the previous 2x2 assembly problem consisting of 43,560 meshes. Because the problem has a strong asymmetry on the radial power distribution, the preconditioned KSMs were excluded in this test. In addition, the performance of the BiCGSTAB is generally better than that of the GMRES as shown in the previous performance tests, the BiCGSTAB is only examined for this problem.

The results of the parallel calculation of the problem are given at Table 5-4 and Figure 5-3. When it is compared with the result of the 2x2 assembly problem, the better parallel efficiency is observed by achieving 8.00 speedup with 12 processors.

The ideal speedup of total computing time is also calculated. Because only the processes of solving linear system are parallelized, the computing time of remaining parts is not reduced regardless of the number of processors. When it is considered that the computing time of solving linear systems takes 68.9% over the total computing time in the serial calculation, the maximum achievable speedup of total computing time is 3.21 in this problem. Parallelization of rest parts is necessary further to increase the parallel efficiency of total computing time of the code.

Table 5-4. Computing time and parallel efficiency of 9x9 mini MSLB problem with unpreconditioned BiCGSTAB

# of thread	Total time (sec)	Cal. time for solving pressure Eq. (sec)	Speedup	Efficiency (%)
1	2,459	1,643	1.00	100.0
2	1,525	746	2.20	110.1
4	1,177	405	4.05	101.3
6	1,060	293	5.61	93.5
8	1,008	239	6.87	85.9
10	980	217	7.58	75.8
12	980	205	8.00	66.7

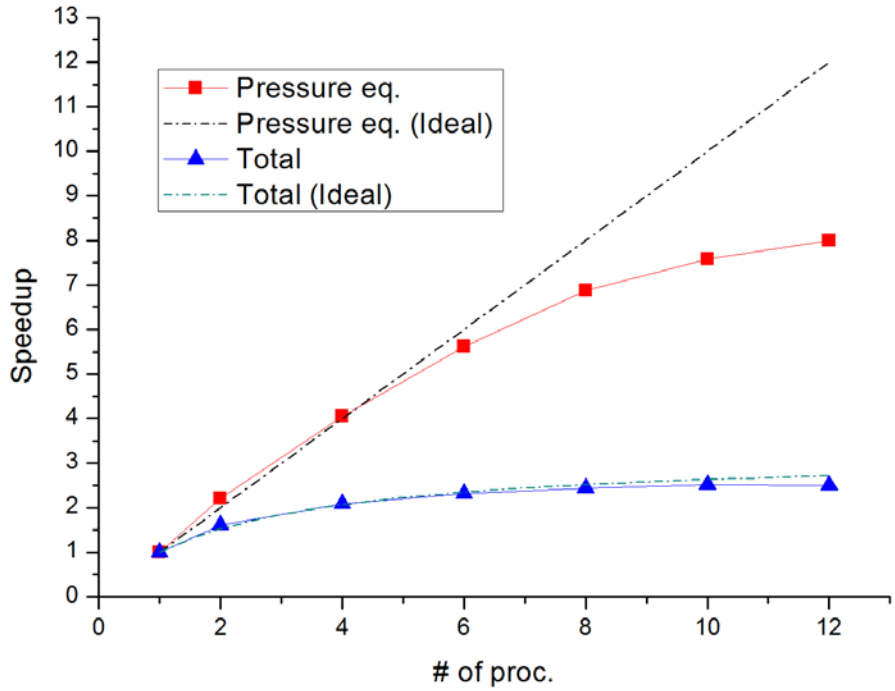

Figure 5-3. Speedup of parallel 9x9 mini MSLB calculation

Chapter 6. Summary and Conclusions

The initial version of the ESCOT pinwise core T/H analysis code was developed based on the drift-flux model and the SIMPLE-like numerical solution scheme. The verification and validation of the code were performed for a set of single-phase flow cases to assure the accuracy. The verification problems include the following phenomena: pressure drop by gravity acceleration, turbulent energy mixing, and friction flow split. In addition, the turbulent momentum mixing and the pressure drop by spacer grids were verified by solving the CNEN 4x4 problem. The solution for the Westinghouse 14x14 problem confirms that ESCOT is capable of analyzing crossflows and reverse flows. For the validation at heated conditions, the mini MSLB-like problem consisting of a 9x9 array of fuel assemblies was simulated. Code-to-code comparisons were carried out using the results of other codes such as CUPID, CTF and MATRA. It turned out that ESCOT is about 3 times faster than CTF while retaining comparable accuracy.

Various linear solvers dealing with linear systems resulted from the SIMPLE algorithm were examined to find the most effective one in the aspect of parallel execution. From the examination results, the following three conclusions are drawn. First, the KSMs are much more efficient than the direct solver. The higher efficiency is obtained for the larger problem. In addition, applying KSMs for momentum equations is powerful due to the well-conditioned momentum matrix. More than 2,000 times speedup was achieved over the direct solver. Secondly, the preconditioning with BILU3D is only effective with GMRES. Moreover, it is turned out that applying BILU3D with the current linear system involving the pressure equation deteriorates the solution performance for the non-uniform power distribution cases. Lastly, the unpreconditioned BiCGSTAB shows the best performance in the aspect of computing time and speedup. That is because the algorithms of unpreconditioned KSMs consist of only multiplication operations of matrix-vector or vector-vector that are naturally parallelizable.

This work demonstrated that a drift-flux model based code can save a certain degree of computing time in single-phase problems, compared to a two-fluid three field based code. In addition, the unpreconditioned BiCGSTAB is the most efficient linear solver for the derived linear system.

The further developments and improvements of ESCOT are necessary. Because the one of the final goals of ESCOT is to make it to be applicable to two-phase flow analysis, V&V for two-phase flow should be performed. Proper models and

correlations of the drift-flux model have to be chosen and examined.

In addition, the parallel performance of unpreconditioned BiCGSTAB is not enough to be applied to solving the whole-core-size problems on massively parallel computers. Two ways of improving the performance are suggested. One way is to increase implicitness of the discretized equations since the current semi-implicit scheme requires excessively many time steps even for null transient situations. The other way is to formulate a steady-state solution scheme to avoid the limit of Courant number. Both ways are related with reducing the number of outer iterations by increasing allowable time step sizes. A better parallel performance is expected, because more computational work is to be performed at each iteration.

Nomenclature

A_c	Flow area of channel (m^2)
C_0	Distribution parameter
D_h	Hydraulic diameter (m)
D_{rod}	Rod diameter (m)
F	Convective term in which the upwind scheme is applied
f	Darcy friction factor
G	Mass flux ($kg\ m^{-2}\ s^{-1}$)
g	Gravitational acceleration ($m\ s^{-2}$)
H	Heat transfer coefficient ($W\ m^{-2}\ K^{-1}$)
h	Specific enthalpy ($J\ kg^{-1}$)
j	Superficial velocity ($m\ s^{-1}$)
K_c	Form loss coefficient
l	Channel-to-channel center distance (m)
$\dot{m}$	Mass flow rate ($kg\ s^{-1}$)
nb	The number of radial neighbors
nnb	The number of total neighbors
P	Pressure (Pa)
P_w	Wetted perimeter (m)
Re	Reynolds number
s	Gap width (m)
T	Temperature (K)

u	Velocity of radial direction (m s^{-1})
V_{gj}	Gas drift velocity (m s^{-1})
v	Velocity of radial direction (m s^{-1})
w	Velocity of axial direction (m s^{-1})

Greek letter

α	Void faction
ρ	Density (kg m^{-3})
ξ_c	Heated perimeter (m)

Subscript

dw	Lower neighbor
ic	Radial neighbor
l	Liquid phase
m	Mixture
sat	Saturated phase
up	Upper neighbor
v	Vapor phase
w	wall

Superscript

n	Time level index

REFERENCE

1. R. SALKO, and M. Avramova, “CTF Theory Manual,” Pennsylvania State University (2014).

2. B. Kochunas et al., “Overview of Development and Design of MPACT: Michigan Parallel Characteristics Transport Code,” Proc. Int. Conf. Mathematics and Computational Methods Applied to Nuclear Science & Engineering (M&C 2013), American Nuclear Society, Sun Valley, ID, USA [CD-ROM] (2013).

3. A. Seubert *et al.*, “The time-dependent 3D discrete ordinates code TORT-TD with thermal-hydraulic feedback by ATHLET models,” PHYSOR 2008, Interlaken, Switzerland (2008).

4. Y. S. Jung *el al*., “Practical Numerical Reactor Employing Direct Whole Core Neutron Transport and Subchannel Thermal/Hydraulics Solvers,” Ann. Nucl. Eng., **62**, 357 (2013).

5. B. Kochunas, “Coupled Single Assembly Solution with CTF/MPACT (Problem 6),” CASL-U-2013-0280-000, Consortium for Advanced Simulation of LWRS (2013).

6. M. Christienne *et al*., “Coupled TORT-TD/CTF Capability for High-Fidelity LWR Core Calculations,” *PHYSOR 2010*, Pittsburg, USA (2010).

7. J. Lee *et al*., “nTRACER/CTF Coupling and Initial Assessment,” *Trans. Kr. Nucl. Soc.*, Jeju, Korea, May 7-8, 2015, Korean Nuclear Society (2015).

8. J. C. Gehin *el al*., “CASL Energy Innovation Hub: Delivering Multi-scale Multiphysics Solutions for Commercial Nuclear Industry Challenges,” ANS Webinar, Consortium for Advanced Simulation of LWRs (2015).

9. D. Hwang et al., “Validation of a Subchannel Analysis Code MATRA Version 1.0,” KAERI/TR-3639/2008, Korea Atomic Energy Research Institute (2008).

10. B. O. Cho *et al*., “MASTER-3.0 Methodology Manual,” KAERI/TR-2061/2002, Korea Atomic Energy Research Institute (2002).

11. J. Y. Cho *et al*., “DeCART v1.2 User’s Manual,” KAERI/TR-3438/2007, Korea Atomic Energy Research Institute (2007).

12. S. J. Kim *et al*., “Preliminary Coupling of MATRA Code for Multi-Physics Analysis,” *Trans. Kr. Nucl. Soc.*, Jeju, Korea, May 29-30, Korean Nuclear Society (2014).

13. C. B. Shim *et al*., “Cross Flow Modeling in Direct Whole Core Transport Calculation with a Subchannel Solver,” *Proc. ICAPP 2013*, Jeju, Korea, Apr.

14-18, (2013) (CD-ROM).

14. N. Zuber, and J. A. Findlay, "Average Volumetric Concentration in Two-phase Flow Systems," *Journal of Heat Transfer*, **87**, pp. 453-468, ISSN 0022-1481 (1965)

15. L. WOLF *el al*, "WOSUB- A Subchannel Code for Steady-state and Transient Thermal-hydraulic Analysis of BWR fuel Pin Bundles, Volume I, Model Description," MIT-EL-78-023 (1978).

16. M. B. CARVER *et al*, "The Drift Flux Model in the ASSERT Subchannel Code," *Nucl. J. CANADA*, **1**(2), 153 (1987).

17. W. WULFF et al, "Development of a Computer Code for Thermal Hydraulics of Reactors (THOR)," BNL-NUREG-50569 (1976).

18. Y. J. Chung *et al.*, "Development and assessment of system analysis code, TASS/SMR for integral reactor, SMART," *Nuclear Engineering and Design*, **244**, pp. 52-60 (2012).

19. D. Maier and P. Coddington, "Evaluation of the slip options in RETRAN-3D," *Nucl. Technol.*, **128**, pp. 153-168 (1999).

20. S. V. Patankar, *Numerical Heat Transfer and Fluid Flow*, Taylor & Francis, ISBN 978-0-89116-522-4 (1980).

21. F. H. Harlow and A. A. Amsden, "A numerical fluid dynamics calculation method for all flow speeds," *J. Comput. Phys.*, **8**, pp. 197-213 (1971).

22. A.A. Amsden and F. H. Harlow, *The SMAC Method: A numerical Technique for Calculating Incompressible Flows*, Los Alamos Scientific Lab. Rept. LA-4370, Los Alamos, New Mexico (1970).

23. T. J. Downar and H. G. Joo, "A preconditioned Krylov method for solution of the multi-dimensional, two fluid hydrodynamics equations," *Ann. of Nuc. Eng.*, **28**, pp. 1251-1267 (2001)

24. M. Mohitpour *et al*., "An advancement in iterative solution schemes for three-dimensional, two-fluid modeling of two-phase flow in PWR fuel bundles," *Ann. of Nuc. Eng.*, **63**, pp. 83-99 (2014)

25. X. S. Li *et al*, "SuperLU Users' Guide," University of California (2016).

26. Y. Saad and M. H. Schultz, "A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems," *SIAM J. Sci. Stat. Comput.*, **7**, pp. 856-869 (1986).

27. H. A. Van Der Vorst, "BI-CGSTAB: A Fast and Smoothly Converging Variant of BI-CG for the Solution of Nonsymmetric Linear Systems," *SIAM J. Sci. Stat. Comput*., **13**, pp. 631-644 (1992).

28. P. Concus *et al.*, “Block Preconditioning for the Conjugate Gradient Method,” *SIAM J. Sci. Stat. Comput.*, **8**, pp. 220-252 (1985).

29. G. Meurant, “Domain Decomposition vs. Block Preconditioning,” Proceedings of the first international symposium on domain decomposition methods for partial differential equations, R. Glowinsky et al., pp. 237-249, SIAM (1988)

30. H. G. Joo, *Domain Decomposition Methods for Nonlinear Nodal Spatial Kinetics*, Ph. D Thesis, Purdue University (1996).

31. S. Mostafa Ghiaasiaan, *Two-Phase Flow, Boiling, and Condensation in Conventional and Miniature Systems*, Ch. 15, Cambridge University Press (2007).

32. T. Ozaki and T. Hibiki, “Drift-flux model for rod bundle geometry,” *Prog. in Nucl. Eng.*, **83**, pp. 229-247 (2015).

33. R. T. Lahey, “Two-phase Flow in Boiling Water Nuclear Reactors,” NEDO 13388 (1974)

34. M. Ishii, “One-dimensional Drift-flux Model and Constitutive Equations for Relative Motion between Phases in Various Two-Phase Flow Regimes,” ANL-77-47, Argonne National Laboratory, Argonne (1977).

35. J. E. Julia *et al.*, “Drift-flux model in a subchannel of rod bundle geometry,” *Int. J. Heat. Mass Transf.* **52**, pp. 3032-3041 (2009).

36. S. W. Chen *et al*., “One-dimensional drift-flux model for two-phase flow in pool rod bundle systems,” *Int. J. Multiph. Flow*, **40**, pp. 166-177 (2012).

37. I. Kataoka and M. Ishii, “Drift-flux model for large diameter pipe and new correlation for pool void fraction,” *Int. J. Heat. Mass Transf.*, **30**, pp. 1927-1939 (1987).

38. T. Hibiki and M. Ishii, “One-dimensional drift-flux model for two-phase flow in a large diameter pipe,” *Int. J. Heat. Mass Transf.*, **46**, pp. 1773-1790 (2003).

39. H. Goda *et al.*, “Drift-flux model for downward two-phase flow,” *Int. J. Heat. Mass Transf.*, **46**, pp. 4835-4844 (2003).

40. C. Clark *et al*., “Drift-flux correlation for rod bundle geometries,” *Int. J. Heat. Fluid Flow*, **48**, (2014).

41. T. Hibiki and M. Ishii, “Distribution parameter and drift velocity of drift-flux model in bubbly flow,” *Int. J. Heat. Mass Transf.*, **45**, pp. 707-721 (2002).

42. T. Hibiki *et al.*, “One-dimensional drift-flux model at reduced gravity conditions,” *AIAA J.*, **44**, pp. 1635-1642 (2006).

43. T. Hibiki *et al.*, "Modeling of bubble-layer thickness for formulation of one-dimensional interfacial area transport equation in subcooled boiling two-phase flow," *Int. J. Heat. Mass Transf.*, **46**, pp. 1409-1423 (2003).

44. J. E. Kelly *et al.*, "THERMIT-2: A Two-fluid Model for Light Water Reactor Subchannel Transient Analysis," MIT-EL-81-014, Massachusetts Institute of Technology (1981).

45. J. T. Rogers and R. G. Rosehart, "Mixing by Turbulent Interchange in Fuel Bundles, Correlations and Inferences," ASME, 72-HT-53 (1972).

46. B. Chexal and G. Lellouche, "The chexal-Lellouche Void Fraction Correlation for Generalized Applications," NSAC-139 (1991).

47. N. E. Todreas, M. S. Kazimi, *Nuclear Systems Volume I: Thermal Hydraulic Fundamentals*, Ch. 11, 2nd ed., CRC Press (2011)

48. Y. J. Chung *et al.*, "Applicability of TASS/SMR using drift flux model for SMART LOCA analysis," *Nucl. Eng. and Design*, **262**, pp. 228-234 (2013)

49. H. K. Versteeg and W. Malalasekera, *An introduction to Computational Fluid Dynamics: The Finite Volume Method*, 2nd ed., Pearson Education, England (2007)

50. D. R. Liles and W. H. Reed, "A semi-implicit method for two-phase fluid dynamics," *J. Comput. Phys.*, **26**, pp. 390-407 (1978).

51. M. Avramova, Development of an Innovative Spacer Grid Model Utilizing Computational Fluid Dynamics within a Subchannel Analysis Tool, Ph. D Thesis, The Pennsylvania State University, U.S. (2007).

52. J. P. Van Doormal, and G. D. Raithby, "Enhancement of the SIMPLE Method for Predicting Incompressible Fluid Flows," *Numerical Heat Transfer*, **7**, 147 (1984).

53. I. E. Barton, "Comparison of SIMPLE- and PISO-type Algorithms for Transient Flows," Int. J. for Numerical Methods in Fluids, **26**, pp. 459-483 (1998)

54. J. J. Jeong *et al.*, "A semi-implicit numerical scheme for a transient two-fluid three-field model on an unstructured grid," *Int. comm. Heat Mass Transfer*, **35**, pp. 597-605 (2008).

55. H. Y. YOON and J.J. Jeong, "A Continuity-Based Semi-Implicit Scheme for Transient Two-phase Flows," *J. of Nucl. Sci. and Tech.*, **47:9**, pp. 779-789 (2010).

56. R. K. Salko *et al.*, "CTF Verification," CASL-U-2015-XXXX-000, The Pennsylvania State University (2015)

57. V. Marinelli *et al.*, "Experimental Investigation on Mass Velocity Distribution

and Velocity Profiles in an LWR Rod Bundle," European Two-Phase Flow Group Meeting, Casaccia, Rome, Italy (1972)

58. H. Kwon *et al*., "Validation of a Subchannel Analysis Code MATRA Version 1.1," KAERI/TR-5581/2014, Korea Atomic Energy Research Institute (2014) (in Korean).

59. S. J. Yoon et al., "CUPID 노심열수력해석 적용 기술 연구," Seoul National University (2016) (in Korean).

60. H. Chelemer *et al*., "THINC-IV: An Improved Program for Thermal Hydraulic Analysis of Rod Bundle Cores," WCAP-7956, Westinghouse Electric Corp. (1973).

61. M. R. Hestenes and E. Stiefel, "Methods of Conjugate Gradients for Solving Linear Systems," *J. Res. Nat. Bur. Standards*, **49**, pp. 409-436 (1952)

62. Y. Saad, Iterative Methods for Sparse Linear Systems, PWS Publishing Company (1996)

63. Ville Hovi, "Calculations of Boiling Two-Phase Flow Using a Porous Media Model," VTT-R-04858-08, VVT (2008).

초 록

본 연구에서는 노심/열수력 연계 다물리 해석 효율성 증진을 위해 Drift-flux 모델과 SIMPLE 계열 수치 알고리즘을 기반으로 하는 봉단위 노심 열수력 해석코드인 ESCOT를 개발하였다. 코드에 구현되어 있는 3차원 4-equation 모델의 지배방정식과 차분 방정식을 유도하였으며, 스칼라 방정식들과 연계된 압력방정식을 유도하였다.

코드의 정확도를 보장하기 위해 단상 유동에 관한 다양한 확인 및 검증 (V&V)을 수행하였다. 확인 및 검증은 해석적 해, 실험값, CTF, MATRA, CUPID 코드 계산값과의 비교를 통해 이루어졌다. 중력가속도 및 지지격자에 의한 압력 강하, 난류 혼합, 마찰흐름분리 및 비대칭 유입에 의한 교차류, 역류, 단순화된 주증기관 파단사고 (MSLB) 현상들에 대해 ESCOT가 유효한 결과값을 계산해 냄을 확인하였다. 또한, CTF 대비 약 3배가량의 계산시간 절감이 가능함을 확인하였다.

병렬계산환경에서 효율적인 선형계 해법모듈을 찾기 위해, 직접 해법 모듈인 SuperLU와 Krylov 부공간 방법인 GMRES와 BiCGSTAB의 효율성을 시험하였다. Krylov 부공간 방법의 preconditioner로서 BILU3D를 적용하였으며, OpenMP를 사용하여 Krylov 부공간 방법의 연산을 병렬화하였다. Preconditioner 방정식의 전향 및 후향 대입 연산 병렬화를 위해 불완전 영역분할법을 도입하였다. 여러 선형계 해법 모듈의 병렬계산 성능 테스트를 수행해본 결과, preconditioning 하지 않은 BiCGSTAB이 계산 시간과 병렬효율 측면에서 가장 우수함을 확인하였다.

주 요 어 : Drift-flux 모델
SIMPLE 알고리즘
부수로해석
Krylov 부공간 방법
병렬계산
학 번 : 2014-20533